LA PRIMA RIVISTA ITALIANA DEDICATA INTERAMENTE AL MONDO LINUX

Linux&C.

www.oltrelinux.com

MENSILE - RELEASE 0.0.0 - OTTOBRE

~~L. 9.000~~ OMAGGIO!

DAL PROSSIMO NUMERO

IN REGALO CD-ROM



IN EDICOLA IL 15 OTTOBRE

DISTRIBUZIONI

Ma quanti sono questi Linux?

Abbiamo deciso di installare Linux. Ed arriva immediatamente il primo problema: quale distribuzione scegliere? Ecco una analisi dettagliata delle più in uso...

HACKERS-MANIA

Basta poco per difendersi dai pirati del WEB

Non navighi sicuro? Hai paura di spiacevoli «abbordaggi»? Niente paura: basta poco per proteggere il tuo sistema Linux dagli attacchi dei malintenzionati.

ED ANCORA...

- **INSECURITY NEWS** Notizie di bugs&patches.
- **INSTALLAZIONE** Linux: ma dove lo installiamo?
- **CONFIGURANDO** Un aspetto diverso per ogni numero: questo mese si parte con **X**, il server grafico.
- **MASTERIZZARE CD** Sotto Linux non è mai stato così facile. Una guida per tutti. Da non perdere.
- **LINUX C/C++** Una presentazione sul compilatore GNU Gcc.
- **...ED ALTRO ANCORA!** XML - il futuro dell'html - Corso di LaTex - lo standard - Hack-It - noi c'eravamo -

LINUX IN AZIENDA

Linux SERVER!

Web Server, Sql Server: la potenza allo stato puro!

LINUX A SCUOLA

Promosso a pieni voti

Linux e' ideale per la scuola: potente, flessibile e senza costi. Una serie di articoli porterà alla sua adozione come soluzione globale.

FreeBSD: l'eroe sconosciuto del Web!

Quale futuro per Linux?

Una nuova rivista (la prima su carta!) su Linux e sull'OpenSource... ma non solo. Fino ad oggi gli appassionati di questo sistema hanno dovuto "rubare" le pagine loro riservate, vagando tra i magazine in edicola. Spero che la loro ricerca sia terminata o, almeno, abbia raggiunto un punto fermo.
Chi si interessa oggi a Linux? Solo studenti, ricercatori ed appassionati? Non proprio, o, meglio, non solo: é grazie a loro che Linux è cresciuto fino a diventare un sistema stabile, performante e competivo. Con grande sforzo e tanta passione.
E adesso che è un sistema con affidabilità e potenza al pari, se non migliore, di altri sistemi ben più conosciuti e blasonati, molte società si stanno indirizzando su di esso, stufe forse di spendere milioni nei cosidetti «*software almost good*».
Linux è una realtà in rapidissima espansione. Grazie alle sue qualità e non ad una politica di marketing sfrenata ed aggressiva. Una garanzia per tutti, utenti e sviluppatori. E la garanzia più grande è costituita dal fatto che i sorgenti del sistema appartengono alla comunità e sono disponibili gratuitamente. I vantaggi sono facilmente comprensibili da chiunque.
Molti (specie chi proviene dal mondo Windows) si chiedono se sono presenti strumenti per sviluppare applicazioni: ce ne sono, e pure di ottimi (e sono anche freeware). Ed altri (commerciali) stanno arrivando: presto si potranno trovare su Linux molti dei software di sviluppo presenti su altre piattaforme: annunciati porting di database e tool di programmazione RAD *in primis*, solo per citarne alcuni. Segno che Linux interessa tutti i vari livelli della scala "informatica".
Alcuni pacchetti non serviranno proprio: Linux, ad esempio, dispone già di server Web tra i più potenti ed apprezzati, Apache e Jagsaw. Sempre con i sorgenti.
Presto si potrà passare a Linux senza cambiare il modo di lavorare attuale: bello, non credete?
Linux è una realtà oggi, e tutto lascia pensare che lo sarà sempre di più in futuro: non conoscerlo, o, peggio, ignorarlo, equivarrà a rimanere ai margini del mercato, rimando legati a tecnologie che saranno obsolete in pochi anni.
Sulle note di un noto gingle pubblicitario, verrebbe da dire: Linux, before it's too late!

Quali argomenti tratteremo in questa nuova rivista?
Tre anime vivono in essa, che interagiranno fortemente tra loro: una riguarderà l'ottimizzazione del sistema, passo obbligato per uno sfruttamento al massimi livelli delle possibilità offerte dal sistema. Un'altra seguirà lo sviluppo software, e quindi proporrà soluzioni che vedono Linux sia come server (SQL server, Mail server, Web Server…), sia come client. L'ultima riguarderà temi generali dell'informatica, riferibili, in ogni caso, al sistema in uso.

Questo numero (lo zero) sarà distribuito omaggio - in foliazione ridotta -, affinché si possa valutare la qualità e lo stile degli articoli. Vi aspetto in edicola il 15 ottobre con il numero 1, che conterrà un cdrom interamente dedicato a Linux ed un simpatico omaggio per voi, un mousepad con il logo del pinguino e il calendario del 2000 (guardate in ultima pagina). Mi raccomando, non mancate.

Linux & C.

Ottobre 1999 - Release 0.0.0

Editore
Edizioni Piscopo

Direttore Responsabile
Patrizio Tassone
(p.tassone@oltrelinux.com)

Grafica ed Impaginazione
Edizioni Vinco
grafica@oltrelinux.com

Hanno Collaborato:
Lorenzo Lazzeri
(l.lazzeri@oltrelinux.com)

Emanuele Busuito
(e.busuito@oltrelinux.com)

Jilani Khaldi
(j.khaldi@oltrelinux.com)

Marco Ivaldi
(m.ivaldi@oltrelinux.com)

Felice Mainolfi
(f.mainolfi@oltrelinux.com)

Filippo Stefanelli
(f.stefanelli@oltrelinux.com)

Riccardo Murri
(r.murri@oltrelinux.com)

Salvatore Sanfilippo
(s.sanfilippo@oltrelinux.com)

Pubblicita
Edizioni Vinco
Tel. 0572 / 91.39.46
Fax. 0572 / 90.10.02
(pubblicita@oltrelinux.com)

per contattare la redazione
Edizioni Vinco
Redazione Linux & C.
Casella Postale 41
51010 Massa e Cozzile - Pistoia
Tel. 0572 - 91.39.46
Fax. 0572 - 90.10.02

http://www.oltrelinux.com
redazione@oltrelinux.com

Stampa
RotoEffe S.p.A. - Roma

Distribuzione
Me.Pe. S.p.A. - Milano



SOMMARIO

Ottobre 1999 Numero 0

Legenda Articoli

BASE

INTERMEDIO

AVANZATO

EXPERT

File Edit View Go Window Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: http://www.oltrelinux.com What's Related

NEWS UPDATE

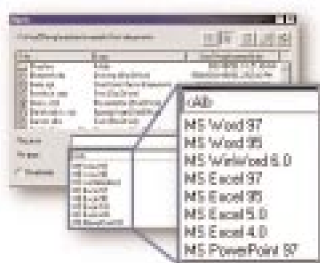


a cura di Marco Ivaldi (raptor)

Sun annuncia StarOffice gratuito

www.stardivision.com

Sun Microsystem ha recentemente acquisito Star Division, una società - fondata 14 anni da Marco Boerries - specializzata nello sviluppo sotto Linux. Come conseguenza di tale acquisizione, Sun ha annunciato ufficialmente il 30 agosto la sua intenzione di distribuire gratuitamente il concorrente di Microsoft Office, chiamato StarOffice, aumentando così la pressione verso una delle sue più grandi concorrenti, la stessa Microsoft. Inoltre Sun afferma che renderà disponibile sotto Sun Community Source License anche il codice sorgente del prodotto. StarOffice è molto simile a Microsoft Office (e con esso molto compatibile), e sono disponibili versioni per Linux, Solaris e Windows. Staroffice - dichiara Sun - ha oltre 4 milioni di utilizzatori. Da parte sua Microsoft non si dice preoccupata da tale decisione ed afferma che non ha in programma alcuna versione della sua suite per Linux.



Microsoft ASP per Linux!

www.chilisoft.com

Chili!soft, leader nel settore delle Active Server Pages (ASP), ha annunciato la sua intenzione di espandere il supporto ASP al sistema operativo Linux. Chili!Soft ASP per Linux permetterà per la prima volta agli sviluppatori di costruire e sviluppare le applicazioni Microsoft ASP sulla più popolare piattaforma Unix degli ultimi tempi. Chili!Soft ASP per Linux sarà disponibile per la fine dell'anno ad un prezzo estremamente contenuto, allo scopo di penetrare il mercato di questo s.o.

IBM premia Linux: VisualAge for Java

www.ibm.com

Finalmente la comunità Linux potrà apprezzare la nuova versione della famosa serie VisualAge: uno strumento RAD (Rapid Application Development) per creare applicazioni java-compatible, applets e JavaBeans. Si possono creare eseguibili in grado di essere lanciate su qualsiasi JVM (Java Virtual Machine) oppure all'interno di un browser java-enabled. Finalmente un ambiente integrato che aumenterà, e non di poco, la produttività degli sviluppatori Linux.

Commercio elettronico in pericolo?

Il 22 Agosto scorso, un team di scienziati guidati da Herman te Riele del CWI di Amsterdam ha trovato i fattori primi di un numero di 512 bit. Questa è la dimensione del 95% delle chiavi crittografiche usate per la protezione del commercio elettronico via Internet: ciò significa che le popolari chiavi RSA di 512 bit non sono più da considerarsi sicure. RSA è un noto sistema crittografico ideato nella metà degli anni settanta da Rivest, Shamir e Adleman presso il Massachusets Institute of Technology di Cambridge. Esso è usato estensivamente nell'hardware e nel software per proteggere il traffico elettronico di dati, come ad esempio nel protocollo SSL (Security Sockets Layer). La quantità di denaro protetta da chiavi di 512 bit è immensa: diversi miliardi di dollari al giorno si spostano



attraverso istituzioni finanziarie in tutto il mondo. Miliardi che presto si troveranno a rischio...

Inprise annuncia Jbuilder per Linux e Solaris

www.inprise.com

JBuilder, il famoso strumento di sviluppo Java, avrà il porting sotto Linux e Solaris: le date di consegna sono previste per fine anno -la versione per Solaris-, per inizio anno prossimo -la versione per Linux-. Jbuilder, uno strumento RAD che permette lo sviluppo per Java, database, business e applicazioni distribuite, sarà il primo prodotto di sviluppo targato Inprise completamente scritto in Java. Sarà distribuito in 3 versioni: Enterprise, Professional e Standard.

Arriva Comache, tool di configurazione multiservizio.

Qual è il modo più semplice di configurare Apache? Utilizzare Comanche, ovviamente! Se un sito è grafico, perché non dovrebbe esserlo il tool per amministrarlo? Ed ecco che, da semplice tool per configurare il più classico dei WebServer, Comanche si è trasformato in un software dall'architettura aperta per la configurazione del Web: il suo codice è stato riscritto, utilizzando al meglio le nuove tecnologie disponibili, quali XML (eXtensible Markup Language), SSL (Secure Socket Layer); presenta una console grafica dalla quale è possibile configurare una varietà di servizi: Samba, Apache e molti altri. Finestre di dialogo, proprietà, possono facilmente essere aggiunte, estendendo così i servizi configurabili in maniera semplice. Un vantaggio non indifferente per gli utenti, che avranno un solo tool di configurazione come riferimento, sia per gli sviluppatori, che potranno concentrarsi sul contenuto del programma di configurazione lasciando all'architettura di Comache la gestione della GUI.

Diskdrake: tool grafico per partizionare il disco rigido.

DiskDrake: è il nuovo prodotto targato MandrakeSoft, la compagnia che sta dietro alla popolare distribuzione Linux Mandrake. Lo scopo di DiskDrake è rendere più facile il partizionamento dell'hard-disk, offrendo una interfaccia grafica, semplice e potente. È scritto interamente in Perl e Perl/GTK, ed è disponibile sotto la licenza GPL (General Public License). Per chi fosse interessato a testare il prodotto: http://www.linux-mandrake.com/diskdrake/

And the winner is...

Al LinuxWorlds Expo è stato tempo di premiazioni: tutti i maggiori aspetti della piattaforma Linux sono stati analizzati e, per ciascuno di questi, sono state date le nomination per i migliori tools. Non sono mancate le critiche per le decisioni della giuria: spesso sono stati esclusi pacchetti molto buoni, semplicemente perché non sufficientemente provati e/o analizzati. Ed andiamo a vedere, a grandi linee, chi è salito sul prestigioso carro dei vincitori: la distribuzione Mandrake un ottimo successo, eletta sia il prodotto dell'anno, sia la migliore distribuzione in ambito server (dove ha preceduto la Debian 2.1). Caldera è risultata la vincitrice in ambito client. Per quanto riguarda l'ambiente grafico, KDE l'ha spuntata su GNOME: in ogni caso questi due ambienti sono presenti entrambi nelle nuove distribuzioni. Interessante notare la conferma della bontà di Apache come WebServer, davvero uno standard in questo tipo di applicazioni. Un'altra conferma da StarOffice come miglior tool di produttività, che supera WordPerfect 8, che paga una minore compatibilità con il mondo Windows. Una grossa diatriba è nata per quanto riguarda il settore database, dove PostGreSQL ha avuto la meglio su Oracle8i. Due ottimi prodotti, anche se molti, Interbase di Borland, Db2 di IBM, avrebbero meritato, se non di vincere, almeno segnalazioni degne di nota. Una conferma anche per le Qt, arrivate alle versione 2.0, davvero ottime librerie per la creazione di applicazioni intuitive e facili da utilizzare (sono free per un uso non commerciale). L'ultimo aspetto che riportiamo è la vittoria di Emacs su Vim, i due storici editor che da sempre dividono la comunità unix su chi, dei due, risulti il migliore…

L'hacker ed il magistrato

Anche quest'anno, nell'ambito di I...ludiamoci '99, la manifestazione ideata dall'Arci di Pescara ed organizzata in collaborazione con l'associazione Metro Olografix tra le altre, avra' luogo l'importante meeting "L'hacker e il magistrato", con conferenze, corsi di base e interventi da parte di noti esponenti del settore. L'appuntamento e' alla stazione FFSS di Pescara dal 1 al 12 settembre 1999.





Linux quale distribuzione scegliere?

Emanuele Busuito
e.busuito@oltrelinux.com

Studente di ingegneria informatica, è un appassionato di sicurezza informatica e programmazione in ambito free UNIX (Linux, *BSD).

Un articolo che analizza le maggiori e più diffuse distribuzioni, evidenziando i PRO e i CONTRO di ciascuna. Una così vasta scelta di prodotti con lo stesso «cuore» ma «optional» diversi permette a chiunque di trovare la versione di Linux che più si adatta alle proprie esigenze.

Linux: abbiamo deciso finalmente di installarlo. Un dilemma: **quale** scegliere? RedHat, Suse, Caldera, Debian, Slackware, Mandrake... ma quanti sono questi Linux? Che differenza c'è tra una distribuzione e l' altra? E cos'è una distribuzione?

Con queste **domande** potrebbe iniziare l' approccio di un nuovo appassionato a Linux.

Sono domande lecite alle quali si possono **aggiungere** le ultime due della serie: **cos'è** in fondo Linux, se **tutti** lo vendono e **tutti** lo personalizzano? Ed allora chi è il **proprietario** di Linux? Vediamo di fare un po' di **chiarezza**.

Linux è il **nome** del **kernel**, ovvero la parte del sistema operativo che si occupa di svolgere le operazioni **fondamentali** come la gestione delle periferiche, delle memoria di massa; il nome deriva da una **combinazione** tra il nome Linus, l' autore, e UNIX, ovvero il tipo di sistema al quale Linux fa riferimento.

Lo sviluppo del kernel Linux è iniziato quasi per gioco da uno studente universitario finlandese (**Linus Torvalds**) che voleva provare la potenza del suo 386 e, al posto di usare un benchmark, decise di scriversi un intero sistema operativo Unix-like (*o almeno così si narra…*): in poco tempo produsse una versione in grado di far funzionare il compilatore GNU Gcc e la shell BASH. L'idea di questo studente attrasse **curiosi** ed **appassionati** che decisero di contribuire al progetto Linux creando drivers per le periferiche più conosciute. Il kernel era funzionante e conteneva il **supporto** per i componenti hardware più comuni: bisognava trovare o sviluppare il software di sistema, gli applicativi, l' interfaccia grafica…: tutto ciò era facilmente reperibile su **internet** dato il gran numero di appassionati che scrivevano software e lo rilascavano sotto licenza GPL (http://www.fsf.org) ovvero completamente gratuito e rendendo disponibili i sorgenti.

Nacquero così le prime **distribuzioni** Linux, ovvero si utilizzava il kernel sviluppato da Linus Torvalds, **corredato** dal software libero disponibile in Internet, e si **confezionava** così un **sistema operativo Unix-like potente e versatile e completamente gratuito**.

Linux - infatti - non è proprietario della società X o Y, ma è di tutti, della **comunità**, chiunque può prendere i sorgenti e modificarli a piacimento e chiunque può confezionare la propria distribuzione senza dover pagare onerose licenze (a patto che si utilizzi solo software freeware).

L'allettante offerta si un sistema operativo con simili caratteristiche, completamente gratuito, attrasse alcune **società** che decisero di investire nel "**fenomeno Linux**" creando le proprie distribuzione in cui usavano il software freeware corredato da alcuni programmi **proprietari** e a cui aggiungevano il supporto **commerciale** (non presente nella distribuzione freeware) creando cosià un sistema interessante per tutti, anche per le **aziende** che, all'epoca, spendevano fior di **milioni** per sistemi unix di **qualità simile** se non inferiore.

Non esiste la distribuzione **adatta** a tutti, fra le varie versioni esistono **differenze** qualitative e quantitative, sul prezzo, sulle piattaforme supportate, sui programmi aggiuntivi forniti, sui metodi di installazione: tutte queste caratteristiche rendono ogni distribuzione **differente**, anche se tutte basate sul solito "**motore**", il kernel Linux appunto.

Nella scelta della distribuzione a noi più adatta dobbiamo considerare molteplici criteri:

- Utilizzo che intendiamo farne (homecomputer, server aziendale, workstation)
- Conoscenza che abbiamo di Linux e dei sistemi UNIX in generale.
- Qualità dell'assistenza della distribuzione.
- Costo della distribuzione.
- Tempo a disposizione per l'installazione e la configurazione.
- Disponibilità della versione in italiano.

Ormai tutte le distribuzioni Linux permettono l'installazione da CDROM (alcune distro usano cdrom boot-able, quindi senza il dischetto di avvio), da NFS (Network File System), FTP, HTTP, harddisk ed alcune persino da floppy (ma ve lo sconsiglio caldamente!): ricordate solo che Linux è un sistema a **32bit** e che quindi non funziona sugli 808x e sui 286.

Le fasi di installazione sono simili in tutte le distribuzioni, in alcune più semplici e in altre più difficili: l'unica parte che rimane abbastanza complessa è il **ripartizionamento** dei dischi che rischia di mandare in fumo il nostro disco (con tutto quello che c'è dentro!): consiglio sempre di effettuare un **backup** almeno dei dati più importanti prima di usare programmi che riducono la dimensione delle partizioni esistenti, che in casi rari possono causare la perdita totale o parziale del contentuto dell' harddisk (è una evenienza davvero molto rara, ma attenzione, in ogni caso!).

Le **principali** distribuzioni Linux sono:

Figura 1: Debian, una delle migliori distribuzioni di Linux, completamente free.

- **Debian** - http://www.debian.org
- **Mandrake** - http://www.linux-mandrake.com
- **OpenLinux** - http://www.caldera.com
- **RedHat** - http://www.redhat.com
- **Slackware**: - http://www.slackware.com
- **SuSE** - http://www.suse.com

Queste non sono le **uniche** distribuzioni, anzi, di distribuzione ne esistono moltissime; in ogni caso queste sono le più utilizzate e famose (in ordine alfabetico):

DEBIAN

http://www.debian.org

Al momento dell'uscita della rivista l'ultima versione rilasciata è la **2.1**. È una distribuzione completamente **gratuita** ed è realizzata da un team di **volontari** sparsi per il mondo con in comune l'hobby per Linux e il free software e un'alta conoscenza dei sistemi UNIX-Like.
È basata sulle moderne **Glibc2** e utilizza il sistema di package proprietario **DEB** (Debian packages) che permette il controllo delle dipendenze dei file e delle librerie e la disinstallazione sicura dei programmi installati, il sistema è molto simile a quello di RedHat (RPM).
Un altro meccanismo interessante per l'installazione dei pacchetti è il comando **apt-get** che permette di scaricare da internet e installare un pacchetto in modo totalmente automatico, rendendo cosi' molto più semplice l'upgrade del sistema.
La distribuzione, nella sua installazione completa, occupa molto spazio e contiene veramente tutto ciò che serve sia all'utente finale sia a chi vuole amministrare un sito internet o un server aziendale.
La debian unisce alla **semplicità d'uso** proprie di altre distribuzioni, la **versatilità** ed una certa **facilità** di **personalizzazione**. Il prezzo da pagare è una installazione piuttosto **lunga**, che però permette di configurare le parti più importanti del sistema.
La distribuzione Debian è tra le più **indicate** per chi vuole usare Linux in modo **professionale** e per chi intende sviluppare software, data la sua ampia disponibilità di programmi precompilati. È anche adatta a chi vuole padroneggiare Linux ed ha molto tempo da dedicarci.

PRO

- Una tre le migliori e più complete distribuzioni Linux.
- Completamente Free.

CONTRO

- Fase di installazione molto lunga e a volte complessa.
- Non compatibile con il formato RPM.
- Assistenza limitata.

MANDRAKE

http://www.linux-mandrake.com

La Mandrake Linux è una distribuzione Linux basata su **RedHat** (con la quale è dichiarata una compatibilità al 99.99%) prodotta dalla Mandrake Soft, giunta alla versione **6.0**.
Questa distribuzione nasce dall'idea di portare **l'intuitività** di altri **sistemi** (Windows in testa) all'interno del sistema Linux, avvantaggiandosi di un ambiente decisamente più **stabile**.
Le caratteristiche più importanti sono quindi la **facilità d'uso** e **semplicità** di **installazione**, grazie ai molti tools presenti: si veda ad esempio il programma incluso per il partizionamento dei dischi (**DiskDrake**), davvero molto comodo da usare grazie alla sua interfaccia grafica.
Questa distribuzione ha **favorito** la diffusione di Linux anche tra gli utenti **meno esperti** per la sua facile amministrazione, grazie anche alla possibilità di configurazione di X-Windows durante la fase di installazione e alla sua partenza automatica attraverso KDM (Kde Display Manager).

Figura 2: La homepage della distribuzione Mandrake in italiano.

Madrake è stata recentemente nominata il "Product of the Year" dal celebre LinuxWorld per le sue caratteristiche di - citando il giudizio dei giudici - flessibilità, prestazioni e facilità di utilizzo.

PRO

- Semplicità d'uso
- Installazione predefinita di KDE già configurato

CONTRO

- Qualche problema nella sicurezza.
- I tool di configurazione nascondono molte parti dell'amministrazione all'utente.

OPENLINUX

http://www.caldera.com

OpenLinux è una distribuzione **commerciale** sviluppata dalla società **Caldera** Inc., è giunta alla versione 2.2 e presenta molti aspetti comuni a RedHat e a SuSE.
Il sistema di package utilizzato è **RPM** che permette installazione/disinstallazione di programmi in assoluta sicurezza e comodità.
Assieme ai programmi **freeware** presenti in tutte le distribuzioni ci sono anche alcuni software **commerciali** degni di nota: SyBase Adaptive Server Enterprise, StarOffice5, WordPerfect 8, e il Caldera NetWare Client.
I programmi proprietari utilizzati per l'amministrazione del sistema sono **LISA** e **COAS**, comodi e semplici da usare.
Questa distribuzione è **particolarmente indicata** nell'ambito **aziendale** per l'ottimo corredo di programmi sia **commerciali** sia **freeware** che rendono più semplice l'utilizzo del sistema e aumentano la produttività.

PRO

- Buon supporto commerciale.
- Corredata da molti software free e a pagamento.

CONTRO

- Non sempre aggiornata come le altre distribuzioni.



Figura 3: Il sito della distribuzione OpenLinux (Caldera).

REDHAT

http://www.redhat.com



Figura 4: Il sito di RedHat, leader *de facto* nelle distribuzioni.

Al momento di uscita della rivista l'ultima versione rilasciata della distribuzione RedHat è la **6.0** che è sviluppata da una società **commerciale** (la RedHat Inc.) che la rende disponibile pure **gratuitamente**: la differenza tra le due versioni è che quella commerciale contiene alcuni software proprietari non presenti nella versione free.
Questa distribuzione è quella che ha **riscosso maggior successo** fra gli utenti, per la sua **semplicità d'uso** e **versatilità** - indubbiamente gli sforzi degli sviluppatori per rendere il sistema più semplice sono ottimi -. L'unico appunto, forse minimo, è che questa ricerca di facilità d'uso **nasconde** alcuni aspetti della gestione del sistema agli utenti finali.
Uno dei punti di forza della RedHat è la presenza del formato file **RPM** (Redhat Package Management) che permette **l'installazione/disinstallazione** dei programmi in assoluta tranquillità con il controllo delle varie dipendenze: nulla ci vieta però di scaricare i sorgenti dei vari programmi e ricompilarli a piacere: per questo esiste il formato proprietario **SRPM** o l'universale **tgz/bz2**. Il formato RPM è liberamente ridistribuibile ed infatti è utilizzato anche da altre **distribuzioni**.
Per installare un programma in formato RPM in un'altra distribuzione esiste il programma **Alien** che si occupa di convertire il formato in quello adatto, a patto che i sistemi di package utilizzati siano presenti nella macchina ove si desidera effettuare l'installazione.

Dalla versione 5.0 la RedHat utilizza le nuove **Glibc2** (GNU C library) al posto delle obsolete **libc5**. Le **libc** sono le librerie dinamiche che vengono richiamate dai programmi in modo da rendere più piccola la dimensione del file eseguibile. Tali librerie contengono le molte **funzioni comuni richieste** dai vari applicativi (gestione files, gestione dei dispositivi, ad esempio) che non avrebbe senso duplicare. Eventuali ottimizzazioni e/o correzioni di bug sono, con questa tecnica, molto più semplici ed efficaci.
Nella versione 6.0 sono presenti i due Desktop enviroment più famosi ovvero **KDE** (http://www.kde.org) e **GNOME** (http://www.gnome.org) che rendono l'utilizzo del sistema molto più semplice e intuitivo attraverso le numerose **utility di configurazione** del sistema e i programmi **applicativi** per i vari servizi internet: web, chat line, newsgroup, mail, icq e molti altri.
Il mio consiglio per chi decide di utilizzare RedHat - dopo ovviamente aver preso dimestichezza con l'ambiente - è quello di non fermarsi solo ai tool di configurazione "che fanno tutto da soli" ma di utilizzare anche la shell (modo terminale!) e lavorare in modalità testo, andando ad **ottimizzare** le varie configurazioni.

PRO

- Facilità d'uso



- Ottimo supporto commerciale.
- Riferimento per la compatibilità di molto software.

CONTRO

- Qualche problema nella sicurezza, almeno nella installazione di default.
- La semplicità d'uso nasconde molte parti dell'amministrazione all'utente.

SLACKWARE

http://www.slackware.com

La Slackware è una delle più **antiche** distribuzioni Linux che ha contribuito al suo successo e diffusione ed è giunta alla versione **4.0** uscita pochi mesi fa, purtroppo si basa ancora sulle obsolete **libc5** ma rimane una delle migliori distribuzioni e più amate dagli "*smanettoni*", assieme alla Debian GNU/Linux. L'installazione è molto **semplificata** rispetto ad altre distribuzioni ma presenta alcuni aspetti **difficoltosi** per gli utenti alle prime armi e forse per questo **non** è molto utilizzata.
La maggior parte dei tool di configurazione sono **testuali** e molto semplici da utilizzare ma la maggior parte degli utenti della Slackware amano configurare il sistema con il classico **editor** di testi (qualunque sia, a proposito esistono vere e proprie guerre su quale sia il migliore).
La Slackware è utilizzata da molti **ISP** per la sua **versatilità** e **potenza** che la rendono ottima anche senza l'installazione di X-Windows e delle utility visuali: la configurazione viene effettuata per lo più da shell testuale, permettendo una **ottimizzazione maggiore, a scapito dell'intuitività**.
Il sistema di package è il classico **tgz** (gzip + tar) che non permette di gestire le dipendenze e si tratta di un semplice sistema di archiviazione: questo è un **grave handicap** rispetto alle altre distribuzioni che hanno potenti sistemi di gestione delle installazioni.
La distribuzione è molto curata anche nell'ambito della **sicurezza** e delle **configurazioni** di default a differenza di altre distribuzioni che presentano **lacune** e dimenticanze in importanti parti del sistema.
La versione commerciale è distribuita da **Walnut Creek** e contiene **6** CD con numerosi mirror di siti di programmi interessanti.

PRO

- Una distribuzione semplice e veloce nell'installazione
- Completamente Free
- Orientata alla sicurezza.

CONTRO

- Usa ancora le libc5, ormai obsolete.
- Poco software incluso nella distribuzione stan¨ dard.
- Non ha un sistema di package per gestire in maniera semplice le installazioni/disinstallazioni.
- Assistenza limitata.

SUSE

http://www.suse.com

La SuSE Linux è prodotta da una ditta tedesca la **SuSE GmBH** che ne fornisce il **supporto** commerciale distribuendo però anche la versione **freeware** in cui **non** sono presenti i software a pagamento.
L'attuale versione rilasciata è la **6.2** basata sulle **Glibc2** e utilizza il sistema di packages **RPM**, ne esiste pure una versione in **italiano** che ne facilita di molto l'utilizzo (a chi non padroneggia l'inglese, almeno!).
La fase di installazione è molto **semplice** specie se si possiede la versione in **italiano** e anche il ripartizionamento dei dischi è più facile rispetto all'utilizzo del comando fdisk.
Per la configurazione del sistema esiste il tool **YaST** che attraverso un interfaccia in modalità testo ci accompagna nella configurazione di tutte le parti del sistema semplificandoci la vita.
La SuSE produce sotto licenza GPL (leggasi gratuito) molti **X-Server** per le più recenti schede video, che vengono poi inserite nelle release di **XFREE86** (X-Windows).
Per la configurazione di X-Windows è presente l'ottimo programma **SaX** che permette di settare con **facilità** le opzioni più **importanti** del file **XF86Config** (il file che contiene tutti i parametri di X-Windows) utilizzando un'interfaccia visuale.



Figura 5: Slackware, una delle più antiche distribuzioni.

PRO

- Buon supporto commerciale.
- Facilità d'uso
- Disponibile in italiano.

CONTRO

- L'utilizzo del tool YAST modifica in modo non standard i file di configurazione, rendendo più difficile la configurazione "manuale".





Figura 6: Il sito della nuova versioneSuSE 6.2 (in italiano).

Jilani Khaldi - j.khaldi@oltrelinux.com

FreeBSD

un eroe SCONOSCIUTO aleggia nel WEB...

Tra le fanfare che accompagnano l'opera di evangelizzazione **Java**, predicata da **Sun** a suono di **miliardi** di **dollari** spesi in marketing, i tamtam dei milioni di sostenitori di **Linux** che vedono in questo sistema operativo un **liberatore** da un **predominio Microsoft** sempre più **asfissiante**, ci sono tanti **piccoli fenomeni informatici** che **avanzano** in **silenzio**, conquistando sempre più consensi, **solo** grazie alle loro intrinseche **qualità**. Tra questi ultimi, **FreeBSD** brilla di **potenza** e di **solidità,** come sistema operativo per il **networking**, che molti sviluppatori e responsabili IT stanno adottando, con grandi **soddisfazioni** e senza il minimo chiasso. Tutti questi fenomeni ed altri, avvengono sotto una fitta nebulosa di nome Internet, che ormai è un **fenomeno** di **massa** e non ha più bisogno di nessuna presentazione.

Perche' FreeBSD?

L'affermazione di Internet come veicolo per le vendite di servizi e beni online, praticabili da chiunque abbia qualcosa da proporre al mercato globale, ha scatenato una febbre contagiosa, simile a quella che ha afflitto le masse di ricercatori d'oro dei secoli precedenti. Internet, non solo favorisce la globalizzazione dell'economia, ma anche quella della cultura e della conoscenza. Basta pensare alle scuole, alle università e agli istituti di ricerca sparsi in tutto il mondo, che possono accedere gratuitamente al Web.
Dietro ad ogni servizio Internet, oppure Intranet (rete locale che usa la tecnologia Internet), c'è sempre un **sistema operativo di rete** che giostra il tutto. Una specie di direttore d'orchestra che controlla ogni bit che si muove all'interno del computer. FreeBSD è uno dei migliori sistemi operativi oggi disponibili per il Web.



Figura 1: Yahoo, uno tra i più famosi motori di ricerca a livello mondiale, utilizza FreeBSD per i propri servizi online.

Fuori le prove

Cito solo due. Yahoo (http://www.yahoo.com) e Hotmail (http://www.hotmail.com) adottano **FreeBSD** su un migliaio di macchine per tutti i loro servizi online su Internet e che ogni giorno servono milioni di utenti sparsi per tutto il mondo. Come ben sapete, Hotmail è stata ultimamente acquisita dal gigante del software Microsoft. Questo la dice lunga sulle **qualità** di FreeBSD per il networking in confronto di tanti sistemi operativi, costosi e blasonati e che da un momento all'altro vanno in crash senza apparenti motivi. A detta dei responsabili di Yahoo.com, prima di adottare FreeBSD, avevano provato vari sistemi operativi, commerciali e free, **senza raggiungere una soddisfazione definitiva**. L'incontro con FreeBSD sembra stato fatale; finalmente, ha messo fine alla loro affannosa ricerca.
E' curioso notare che tutti gli utenti FreeBSD, almeno quelli che ho avuto modo di conoscere, provengono dal mondo Linux. Tutti dicono di aver trovato in FreeBSD le stesse qualità di Linux ed altre ancora. (Vedi tabella di confronto)

Un po' di storia: nobili origini

Molti pensano, che almeno in parte, il successo e la popolarità di Linux sono dovute alla presenza di una figura carismatica dietro tale sistema operativo. **Linus Torvalds**, il giovane studente universitario dell'università di Helsinki che sviluppò **per primo** Linux e continua tuttora a farlo con il contributo, tramite Internet, di molti volenterosi programmatori sparsi per tutto il mondo. Questa tesi è stata confermata da un precedente simile, non molto lontano. La simpatia nei confronti di **Bill Gates** all'epoca della comparsa del PC (inizio anni 80). Il grande Bill era visto dalla comunità dei programmatori come il **paladino della libertà**, che lotta contro il predomino dell'IBM. Bisognava sostenerlo. Oggi, agli occhi di molti, Bill Gates è una specie di cerbero che aggredisce chiunque si avvicini al mercato software, **freeware** compreso, e, con tanto cinismo, cerca solo di difendere e consolidare il suo impero, facendo fatica ad accontentarsi solo del 80% del mercato mondiale del software. L'obiettivo dei due protagonisti è praticamente lo stesso. Dominare il mondo. Con una piccola differenza di natura filosofica ed operativa. Linus sostiene che il software deve essere **disponibile per tutti, gratuitamente e con codice sorgente**. E' una **proprietà intellettuale** che appartiene a tutta l'umanità. Come le **quattro equazioni di Maxwell** e le **leggi di Newton**. Una visione leggermente diversa e contrastata da quella di Gates. Bill vuole tutto per sé.

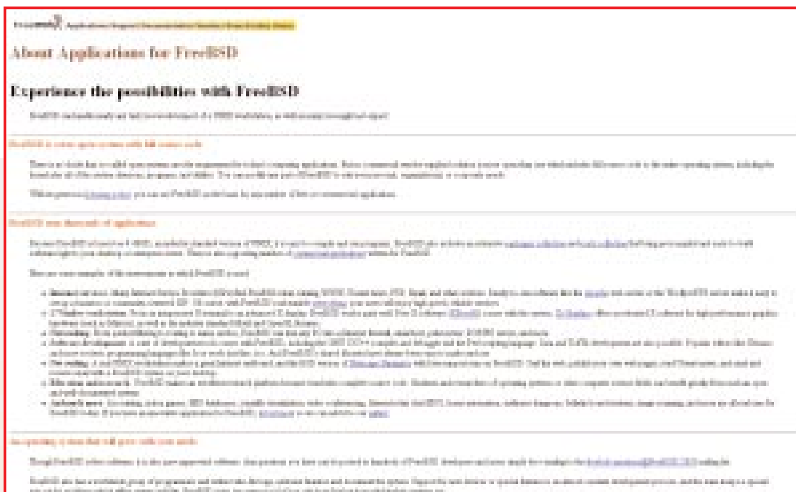


Figura 3: FreeBSD può far girare tantissime applicazioni, in tutti i settori, dalla grafica al networking, da internet ai database.

Dunque, dietro a FreeBSD non c'è nessuna figura carismatica, ma una **istituzione**, e una manciata di scienziati. L'Università di Berkeley della California e una équipe di informatici. Essi continuano con tanto amore il loro tradizionale lavoro certosino di sviluppo e manutenzione di tale sistema operativo, iniziato più di 20 anni fa. Questo spiega le professionali **qualità** e la proverbiale **solidità** di FreeBSD.

UNIX fu inventato all'inizio degli anni 70 e aveva come obiettivo di diventare il sistema operativo universale (unico). Quando uscì il primo PC IBM nel 1981, UNIX era già considerato un sistema operativo maturo. A metà degli anni 80 esistevano ben quattro diverse versioni di UNIX. La Research Version usata solo da AT&T (il gigante americano delle telecomunicazioni); la Berkeley Software Development (BSD), usata dall'università californiana; System V, la versione commerciale della AT&T; ed infine, Santa Cruz Operation o semplicemente SCO.

Al Computer System Research Group (CSRG) dell'università della California fu affidato lo sviluppo di Berkeley UNIX ed era considerato un oggetto di ricerca. Ma ben presto si resero conto che le sue qualità gli permettevano di essere impiegato all'interno delle industrie e nei processi produttivi aziendali. Il CSRG fu sciolto in seguito a tale decisione.

> «Dietro a FreeBSD non c'è nessuna figura carismatica, ma una istituzione e una manciata di scienziati»

Un gruppo di scienziati, dell'ex CSRG, decisero il porting di tale sistema operativo (BSD UNIX) su piattaforma PC, ma nel rispetto della tradizione Berkeley: distribuire il sistema operativo con codice sorgente e senza nessuna restrizione per l'utilizzo ed eventuali modifiche. E così fecero. La reazione della AT&T fu violenta ed accusò il BSD di violazione dei termini della licenza a loro concessa. La AT&T fece causa all'università di Berkeley e la costrinse ad eliminare dalla distribuzione BSD tutto il codice da essa sviluppato.

Bisognava aspettare il **1994** per vedere uscire la prima versione di FreeBSD, derivata da BSD 4.4 Lite, nel totale rispetto della licenza, **sostituendo il suo codice proprietario della AT&T con uno nuovo, sviluppato dal gruppo BSD.**

Ma ormai era troppo tardi. Linux e tutti i sistemi operativi commerciali UNIX erano già in piazza da un pezzo. Microsoft consapevole della posta in gioco, dopo aver conquistato il mercato desktop con Windows, mettendo fuori causa alleati e concorrenti, si stava muovendo per conquistare il server con Windows NT. Impiegando la sua schiacciante potenza economica con investimenti faraonici in marketing e una serie di acquisizioni, secondo il suo primo teorema: **"se non potete battere il vostro rivale fatelo alleato"**. Seguirà il secondo teorema, altrettanto efficace ed attualmente in fase di applicazione: "dal client al server, per tutte le soluzioni software e middleware, usare solo prodotti Microsoft. In questo modo non ci saranno vie di uscita per nessuno. Amen."

Passando dalle preghiere ai fatti, i risultati sono più che evidenti. Oggi, NT sta penetrando con successo come server nelle aziende, senza incontrare grandi ostacoli. La ragione mi pare molto evidente. Il mercato ha sempre dimostrato di essere più sensibile ad ascoltare e premiare chi **promette** e **grida più forte** e non chi dimostra di possedere le qualità migliori. Anche a costo zero. FreeBSD docet.

> «Il mercato ha sempre dimostrato di essere più sensibile ad ascoltare chi promette e grida più forte e non chi posside le qualità migliori»

Installazione: una unica procedura

Una delle maggiori ragioni del successo di Windows è la sua facilissima procedura di Installazione. Potrebbe essere effettuata anche dalla segretaria. Lo stesso, purtroppo, non si può dire dei sistemi UNIX. L'installazione di Linux, per esempio, potrebbe rivelarsi per chiunque una **autentica impresa**. Laureati in informatica compresi. Un altro problema che affligge gli utenti Linux, sono le varie distribuzioni. Ognuna con la propria procedura di installazione. Personalmente, ho provato ben sette distribuzioni! Nessuna era uguale all'altra. Come primo vantaggio, gli **utenti FreeBSD sono esenti da questi incubi**.

Quando si parla di FreeBSD, si parla solo di **versione** e mai di **distribuzione**. Lo sviluppo centralizzato di tutto il sistema operativo e non solo del kernel, come avviene in Linux, ha condotto a dotare FreeBSD di una unica procedura di installazione per tutto il sistema. Molto semplice e efficace. Il tempo necessario per una installazione minima è di circa mezz'ora. Per il resto, tutto dipende dalle applicazioni che l'utente vuole installare. FreeBSD ha un solo comando per installare i programmi e un altro per rimuoverli. Molto pratico.

Applicazioni

FreeBSD fa girare il lunghissimo elenco di programmi OpenSource (ricordiamo che la maggior parte di questi programmi è **stata proprio sviluppata sotto BSD**) che conta più di **2300** applicazioni e sono in aumento ogni giorno. Quasi la totalità dei programmi compilati per Linux nel nuovo formato **ELF** e varie applicazioni commerciali possono essere adoperati senza grossi problemi. **Interbase 4.0** è stato portato ultimamente su FreeBSD ed anche **StarOffice 5.0**. Molti altri



Figura 2: Il simpatico demone ed i logo che testimoniano l'utilizzo del FreeBSD.

Spazio BSD

prodotti sono stati annunciati.
I classici sono sempre di serie: vari Shell, EMACS, **Apache**, sendmail, qmail, strumenti di sviluppo **GNU C/C++**, **Pascal**, SmallTalk, Objective C, Modula 2/3 ed altri, **PHP**, PERL, Python, JDK, **MySQL**, **Postgres**, **Netscape** Navigator, Amaya, ...

FreeBSD vs. Linux: stessa razza, stessa faccia

FreeBSD è visto dai suoi utenti, con grande orgoglio, come il vero UNIX e Linux come un clone di UNIX. Etichette a parte, i due sistemi hanno caratteristiche molto simili e sono compatibili tra loro, grazie alla loro aderenza alle specifiche POSIX. L'unica grande differenza è che Linux è una derivazione di System V, il secondo ramo dell'albero UNIX; il primo è BSD. Un fatto degno di nota è che Linux, mentre aderisce alle specifiche System V, non usa l'interfaccia **Transport Layer Interface** (TLI) per il networking, ma quella dei sockets BSD, **meno evoluti ma molto più efficienti**.
A questo punto, possiamo tranquillamente dire che, sostanzialmente, le differenze sono più nominali che effettive.
Per il networking FreeBSD è **leggermente più performante** di Linux e lo supera almeno del 10-20% confrontando le ultime versioni dei kernel (2.2 di Linux, 3.0 di FreeBSD).

Da Microsoft con affetto: basta pagare

Immaginate se ogni anno, o solo ogni sei mesi, siete chiamati ad aggiornare il vostro frigorifero e il vostro tostapane. E' una prassi semplicemente **illogica ed inaccettabile**. Questa stessa prassi viene **accettata** con una **tranquillità** d'animo quasi olimpica dagli utenti dei PC. Ormai il processo sembra inarrestabile e si muove in due direzioni strettamente legate tra loro. Software sempre più ingombrante e hardware sempre più possente per farlo funzionare. Tutto a carico dell'utente. Ovviamente.
Ma quando finirà questo circolo vizioso? La risposta appare del tutto semplice. Quando si farà uso del buon senso.
Il software commerciale ci ha portato a questa situazione, usando i metodi di marketing più sofisticati, creando la falsa necessità di aggiornarsi in continuazione.
Una delle ricette anti-marketing e contro i falsi bisogni tuttora valida, sembra quella suggerita da Roosevelt: **"usate quel che avete, finché potete, dove siete "**.

Software almost good

Un grande problema dei **sistemi Windows** e della maggior parte dei programmi che girano su di essi, i così certificati Designed for Windows, sono i **continui ed imprevedibili crash**.
L'attributo che viene dato a tali programmi non è **"software bacato"**, come uno onestamente,dovrebbe aspettarsi, ma piuttosto uno più pittoresco **"software almost good"**. E' più bello detto in questo modo. Sembra più perdonabile quando va in crash. Letteralmente significa **"software quasi buono"**. Che in parole ancora più chiare, significa **"software che funziona e non funziona"**.
Immaginate una macchina esposta per la vendita che porta questo cartello: "automobile che cammina e non cammina". Difficilmente troverà un acquirente. Ma se ci mettiamo sopra un marchio di una software house molto nota, potrebbe diventare anche una macchina di **grande successo**.
Ma dobbiamo ancora continuare ad accettare con filosofia un software costoso ed ingombrante, che a volte funziona, altre no? Macché.
Confrontate questa **infelice filosofia** con quella adoperata dal gruppo BSD.
Ogni nuova versione di FreeBSD viene messa a disposizione del pubblico, solo quando i suoi sviluppatori sono sicuri che il **software funziona perfettamente**, e che non ci siano **problemi di stabilità per l'intero sistema**. Tutto ovviamente a costo zero e compreso di codice sorgente.
Anche se i numeri confermano la supremazia della prima, e la sua accettabilità da un **mercato ipnotizzato**, i tempi sembrano maturi per fermarsi e riflettere sul da fare per non rimanere intrappolati in una morsa dalla quale sembra molto difficile liberarsi in seguito.
Comunque, tra le due filosofie, lascio la scelta a voi.

Sul prossimo numero...

Installazione e configurazione: questo troverete nella prossima puntata. Quindi occhio a non perderla. A presto.



Riferimenti

http://www.freebsd.org



FreeBSD	Linux
FreeBSD è un sistema operativo completo, mantenuto da un singolo gruppo di sviluppatori. C'è una sola distribuzione FreeBSD.	Linux è solo un kernel. I programmi e le utilità di sistema che non fanno parte del kernel vengono aggiunti dalle varie distribuzioni.
La stabilità dell'ambiente è la caratteristica più tenuta in considerazione per i futuri sviluppi.	C'è sempre più enfasi a rendere Linux un sistema più rivolto verso la produttività.
La distribuzione unica di FreeBSD lo rende facile da installare ed usare.	Ogni distribuzione ha la propria procedura di installazione con personalizzazione di alcune funzionalità per la gestione del sistema.
La licenza BSD è molto permissiva.	La licenza GPL è meno permissiva di quella BSD

Tabella 1: Breve comparazione tra FreeBSD e Linux



HACKERS-MANIA

paura dei PIRATI?



MARCO IVALDI
m.ivaldi@oltrelinux.com

Si occupa di sicurezza in ambito Unix. Collabora con un ISP nell'amministrazione di un sito Web. È webmaster di dislessici.org.

Questo articolo è diretto a tutti gli utenti "casalinghi" di linux; in esso sono trattate le basi della sicurezza di un sistema Unix. È un articolo introduttivo, che sarà ripreso ed ampliato nei vari aspetti sui prossimi numeri. Già con queste piccole modifiche, in ogni caso, si riesce a raggiungere un livello di sicurezza di tutto rispetto per un uso "normale".

Ok, tutti vi parlano di linux e alla fine anche voi vi siete convinti a provarlo. Dopo una più o meno faticosa prima installazione, finalmente siete davanti alla vostra linux box nuova fiammante. Purtroppo le configurazioni di default spesso non sono proprio il massimo e, quel che e' peggio, possono lasciare dei "buchi" in cui qualche "cattivo ragazzo" della rete può decidere di infilarsi... Che fare quindi per rendere più sicuro in **10** minuti il proprio linux neo-installato? Quelli che seguono sono dei consigli pratici per raggiungere un livello di protezione accettabile per un pc di casa.

Il tuo Linux sicuro in pochi semplici passi!

Password.

Generalmente, indipendentemente dalla distribuzione, la prima scelta a cui ci si trova di fronte durante il "duro lavoro" della configurazione è proprio quella delle **password**. E' molto importante **resistere** alla **tentazione** di usare parole semplici (e per questo facili da ricordare), in quanto questo faciliterebbe notevolmente il lavoro ad un ipotetico **cracker**. Senza entrare appieno nella questione (ma questo potrebbe essere lo spunto per un futuro articolo), Linux e Unix in generale utilizzano un sistema di cifratura delle password chiamato **DES (Data Encryption Standard)**. Questo algoritmo è a "senso unico", non è cioè possibile risalire dalla password criptata a quella in chiaro. Esistono però dei **programmi** in grado di "**indovinare**" le password più semplici relativamente in fretta, basandosi su dei **dizionari** facilmente reperibili in Rete. È quindi buona norma scegliere delle parole senza un senso compiuto, che contengano lettere maiuscole e minuscole, numeri e caratteri speciali. Un'altra cosa fondamentale è l'abilitazione del sistema di ***password shadowing***, che non permette agli utenti normali di leggere le password cifrate. Per verificare velocemente se il vostro sistema usa le shadow passwords:

```
cat /etc/passwd
```

A seconda dei casi potrete ottenere in output delle righe di due generi:

Caso A

```
raptor:DlCrk/.3gm/ud:1000:100:Johnny
Rotten,,,:/home/raptor:/bin/bash
```

oppure

Caso B

```
raptor:x:1000:100:Johnny
Rotten,,,:/home/raptor:/bin/bash
```

A) Se il secondo campo del password file (quello dopo l'userid) contiene dei simboli (la password criptata) come nel primo esempio allora dovete abilitare il supporto utilizzando i comandi pwconv(8) e shadowconfig(8), e modificando il file /etc/login.defs dove richiesto. In questo modo le password cifrate saranno conservate in /etc/shadow, leggibile solo da root.

B) Se invece vedete solo un simbolo (una "x" nell'esempio) state già utilizzando le password shadow di default e dovete solamente preoccuparvi di mantenere il file /etc/shadow con owner root e permessi **700** (lettura/scrittura solo a root), mentre /etc/passwd deve essere leggibile da tutti (e non scrivibile, ovviamente!).
Nelle ultime versioni di tutte le distribuzioni comuni di Linux il supporto per le shadow passwords è **abilitato di default**, quindi non avrete bisogno di fare nulla a parte assicurarvi che i permessi dei file siano corretti.

Pagine del manuale consigliate:

crypt(3) - funzione di password encryption
passwd(1) - comando passwd
passwd(5) - password file /etc/passwd



Figura 1: http://neworder.box.sk, uno dei più conosciuti siti di hacking presenti nella Rete.



shadow(5) - *shadow file /etc/shadow*
pwconv(8) - *comandi pwconv, pwunconv, grpconv, grpunconv*
shadowconfig(8) - *attiva/disattiva le shadow passwords*
login.defs(5) - *file /etc/login.defs*
chmod(1) - *comando chmod per gestire i permessi dei file*
chown(1), chgrp(1) - *per cambiare owner e gruppo dei file*

Servizi: inetd e i file di avvio.

Ora veniamo alla configurazione di rete vera e propria. Linux, come ogni sistema Unix, offre dei servizi per lo più attivi su porte standard. Ad es. abbiamo il **telnet** sulla 23, **l'ftp** sulla 21, il **sendmail** sulla 25 e così via. Molto spesso la configurazione di default attiva gran parte di questi servizi, molti dei quali **non interessano** all'utente medio. Per vedere quali servizi sono attualmente attivi sulla propria macchina utilizzate il comando:

```
netstat -a
```

che mostra in output tutti i socket aperti, compresi quelli "in ascolto" (che sono quelli che interessano a noi).
Ora, meno servizi abbiamo attivi meno sono le possibilità che un ipotetico cracker ne sfrutti le debolezze per attaccare la nostra macchina. Bisogna infatti tenere presente che la **sicurezza** è un **compromesso**: se vogliamo avere un computer **totalmente** sicuro, basta **tenerlo isolato** dalla rete, senza servizi attivi e magari spento. Allora sì non ci sarà da preoccuparsi di nulla, ma il computer **cesserà** anche di essere di qualche **utilità**. Quindi un amministratore di sistema si trova costantemente in **bilico** tra i servizi che vuole offrire (**dns**, **sendmail**, **pop3**...) e le vulnerabilità che essi possono presentare. Bisogna perciò trovare il giusto equilibrio.
Tornando a noi, penso che per un sistema linux "casalingo" tutto ciò non sia particolarmente problematico: in generale consiglio di chiudere inizialmente tutti i servizi, per poi riaprirli in caso di necessità dopo averli attentamente configurati.
In pratica questo si riduce a:

1) commentare tutte le righe del file /etc/inetd.conf, ponendo un "#" all'inizio di ogni riga, **rehashando** poi il demone **inet** con il comando:

```
killall -HUP inetd
```

In questo modo **inetd** (l'**internet daemon**, che si occupa di lanciare i demoni richiesti) **rilegge il file di configurazione** da noi modificato per non avviare alcun servizio e agisce di conseguenza. In caso poi volessimo attivare un servizio in particolare basterà andare a scommentare la riga corrispondente e rehashare nuovamente il demone.

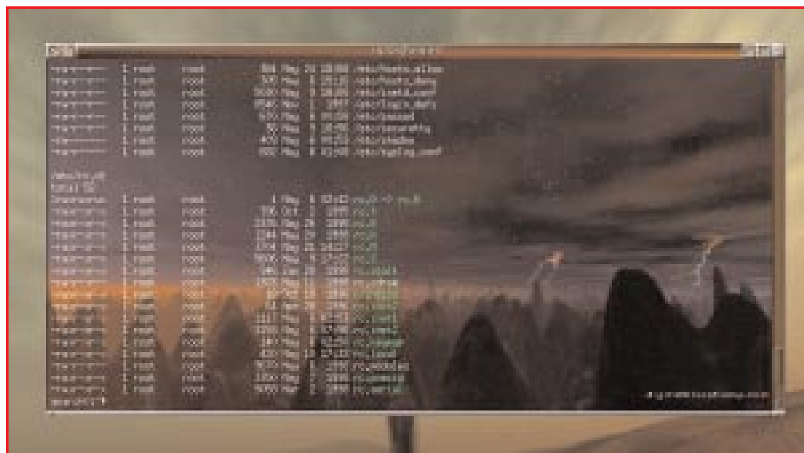


Figura 3: Un elenco dei file della directory /etc/rc.d

2) modificare i file di avvio in modo da non far partire al boot i demoni che non sono attivati da inetd (httpd, smbd, sshd...). A seconda della distribuzione il procedimento varia leggermente: vi rimando alle man pages di init e initscript. In generale, comunque, i file utilizzati da init si trovano in /etc/rc.d o semplicemente /etc.

Commentate anche le parti riguardanti questi ultimi demoni e al prossimo reboot essi non verranno più riavviati. Anche qui, in caso ne abbiate bisogno basterà scommentare le parti in questione. Per ora killate i processi in questo modo:

```
killall -9 [nomedeldemone]
```

Potete verificare con il solito netstat -a quali porte avete chiuso e quali no.

Pagine del manuale consigliate:

inetd(8) - *internet daemon*
netstat(8) - *comando netstat*
init(8) - *inizializzazione del controllo dei processi*
initscript(5) - *script che esegue i comandi di /etc/inittab*
ps(1) - *comando ps che mostra i processi attivi*
kill(1) - *comando kill*
killall(1) - *comando killall*



Figura 2: Un possibile output a video del comando netstat lanciato da xterm.



Tcp wrapper: hosts.allow e hosts.deny.

Come **misura di sicurezza** molti servizi di rete sono gestiti da un **programma** protettivo detto **TCP wrapper** (**tcpd**). I servizi protetti sono quelli elencati in **/etc/inetd.conf** che specificano **/usr/sbin/tcpd**. In pratica, inetd, quando riceve una **connessione** ad una **porta associata** ad un certo **servizio**, provvede a **chiamare tcpd** che a sua volta avvia il **demone** che si occupa del **servizio in questione**, come si vede chiaramente in inetd.conf. In realtà prima di avviarlo tcpd procede a registrare nei **log** la richiesta e fa alcuni **controlli** addizionali (consultare le rispettive man pages per i dettagli).

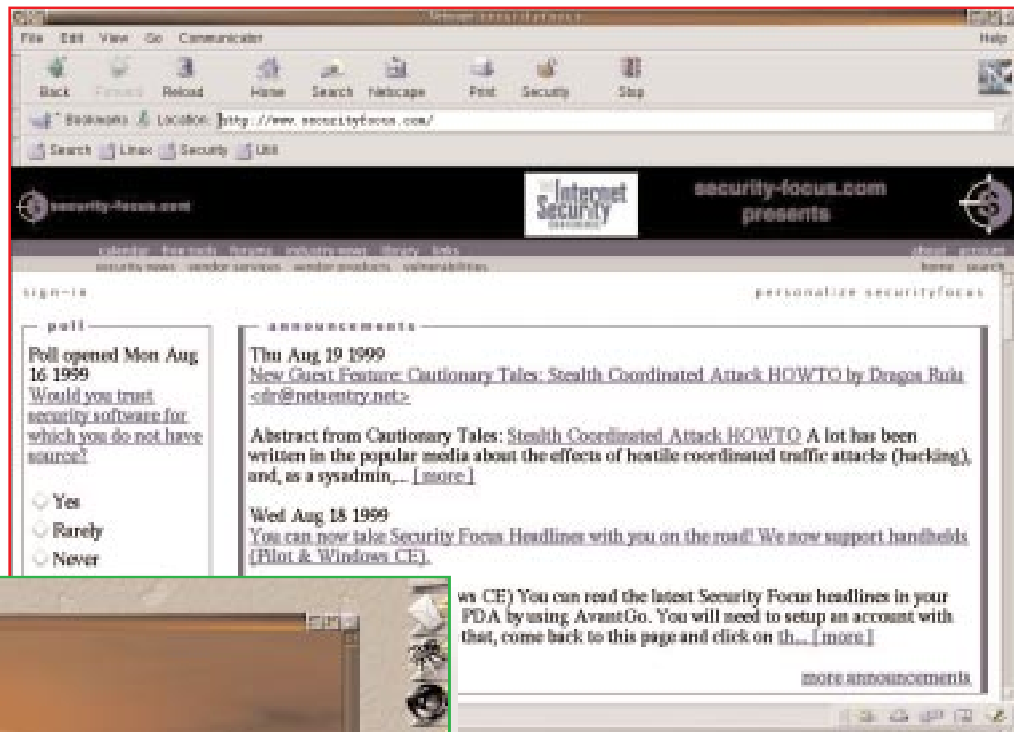


Figura 4: BugTraq, il sito della piu' importante mailing list sulla sicurezza.



Figura 5: Il contenuto del file /etc/syslog.conf

Quello che a noi interessa è la possibilità che tcpd introduce di **discriminare** l'avvio o meno di un servizio in base all'host che ne fa richiesta. In pratica è possibile rendere **disponibili** alcuni servizi solo ad alcuni host ***"fidati"***, ad esempio quelli all'interno della propria stessa rete o addirittura solo al localhost. Per fare cio' si utilizzano i due file di configurazione **/etc/hosts.allow** e **/etc/hosts.deny**. Il primo determina quali host o reti possono **accedere** ad una certa porta, il secondo quali ne sono **esclusi**. La sintassi in entrambi i file è la medesima:

```
servizio : host o subnet [ : comandi di shell opzionali ]
```

dove i vari campi rappresentano:

servizio: nome del demone come da inetd.conf, ad es. "wu.ftpd".

host o subnet: host ai quali si permette (hosts.allow) o non si permette (hosts.deny) l'accesso al servizio indicato.

Esempio

```
127.0.0.1 # specificazione di
                          un host
127.0.0.  # specificazione di
                        una subnet
```

Per dettagli sui comandi di shell del terzo campo vi rimando al manuale. Sono inoltre disponibili delle wildcard, tra cui le più importanti sono:

ALL : tutti gli host (compreso localhost) o tutti i demoni, a seconda del campo in cui si trova.

LOCAL: tutti gli host che non contengono un "." nel nome.

PARANOID: tutti gli host il cui nome non corrisponde al loro indirizzo IP. Per poter utilizzare questa wildcard il tcpd deve essere stato compilato con l'opzione **-DPARANOID**.

Il mio **consiglio** è quello di configurarli nel seguente modo:

hosts.deny: deve contenere semplicemente la riga

```
ALL : ALL
```

che nega l'accesso a tutti per tutti i demoni.

hosts.allow: a seconda delle esigenze.

Generalmente per un computer di casa è buona regola avere una cosa del genere:

```
ALL : localhost   # abilita
          l'accesso a tutti i
         servizi da localhost
```

Potete aggiungere gli altri servizi che desiderate e autorizzarne l'uso agli host voluti.

Pagine del manuale consigliate:

tcpd(8) - TCP wrapper
hosts_access(5) - sintassi dei file hosts.allow e hosts.deny

Consigli generali.

Nell'installare un demone tenete presente che è **sempre** bene avere l'**ultima versione**: in questo modo eventuali **bug** risulteranno corretti, sia per quanto riguarda l'aspetto della **sicurezza** sia per le **prestazioni** del demone. Utilizzate **sempre** il controllo offerto da hosts.deny e hosts.allow, leggete sempre la documentazione e le FAQ relative e, se possibile, tenetevi sempre aggiornati sulle ultime notizie relative alla sicurezza *(ad esempio. inscrivetevi alla mailing list bugtraq - http://www.securityfocus.com/)*. Questo come regola generale.
Ora alcuni suggerimenti molto ***pratici***:

1) Modificate il file /etc/securetty in modo che contenga solamente le tty locali (quelle del tipo ttyN, dove N è un numero, corrispondente al numero dei terminali virtuali). Questo file in pratica contiene l'**elenco** delle **tty** da cui root si può **loggare**: limitandole, si riducono i rischi.

2) Ridirigete l'output del syslog su una tty non utilizzata. Basta aggiungere a /etc/syslog.conf la riga seguente (per la tty10):

```
*.*           /dev/tty10
```

In questo modo premendo ALT+F10 avremo davanti i log aggiornati in tempo reale, senza aver bisogno di andare a guardare i file in /var/log.

3) Leggete con attenzione le man pages riguardanti il syslogd, il demone che si occupa di loggare le attività che hanno luogo sulla macchina. Anche questo potrebbe essere uno spunto per un articolo futuro.

4) Prima di creare degli account per "esterni" sulla vostra macchina assicuratevi di sapere cosa state facendo, soprattutto se avete dei dati "riservati" sul vostro computer.

Pagine del manuale consigliate:

securetty(5) - file /etc/securetty
syslog.conf(5) - file /etc/syslog.conf
syslogd(8) - demone syslogd

Tools utili.

Potreste considerare di installare alcuni tool utili, che migliorano e facilitano il compito di mantenere sicura la propria linux box. Ne nomino alcuni qui di seguito, con la possibilità di un futuro approfondimento.

ipfwadm/ipchains: **firewall** per linux, rispettivamente per kernel 2.0.x e 2.2.x. Richiedono l'abilitazione del supporto nel kernel. Permettono di filtrare agevolmente il traffico sulla propria macchina, basandosi su "rules".
iplogger: set di programmi che **registrano** le connessioni TCP, UDP e ICMP in arrivo, passando l'output al syslog. Molto utile per avere dei log dettagliati.
colorlogs: semplice script in perl che permette di colorare l'output del syslogd onde consentire una lettura più agevole e immediata.
ssh: la secure shell. Client e server per comunicazioni remote criptate.
ttysnoops: tool per "sovrapporsi" alle sessioni degli utenti sulla propria macchina linux e poter così controllare le loro attività. Da usarsi solo per scopi "etici"...
cfsd: file system crittografato, per proteggere i propri dati riservati.
nmap e altri scanner: programmi utili per testare le vulnerabilità della propria macchina.



Figura 3: Il sito www.dislessici.org

Sono poi disponibili moltissimi altri programmi dalle svariate funzioni, reperibili in rete con facilità. Vi consiglio di visitare spesso http://freshmeat.net e il "nostro" http://www.dislessici.org per le ultime novità in questo campo. Per ora, buon divertimento col vostro nuovo Linux!

Sul prossimo numero...

Vedremo come utilizzare il Linux come firewall per aumentare la sicurezza. Non mancate.



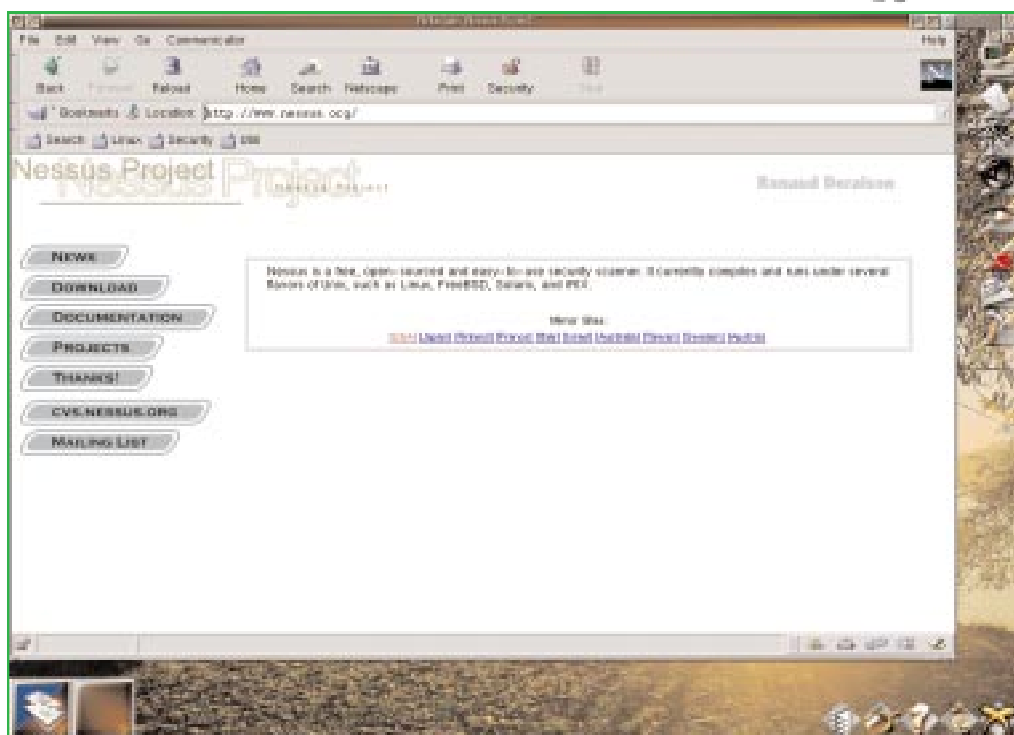
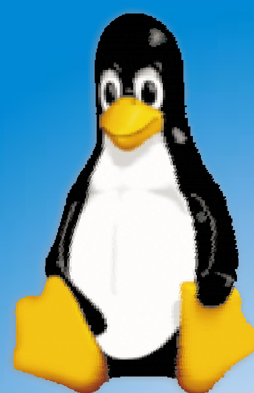
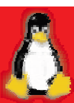


Figura 7: Nessus, uno degli scanner più apprezzati, free e con i sorgenti!



Linux
The World Is Changing
The Operating System That Is



InSicurezza: bugs & patches

INSECURITY NEWS

a cura di antirez

E' possibile reperire le patch alle seguenti URL:

Debian http://www.debian.org/security/
RedHat http://www.redhat.com/corp/support/errata/
SuSE http://www.suse.de/security/
Caldera http://www.calderasystems.com/news/security/

In tutti gli altri casi e' necessario scaricare l'ultima versione della applicazione dal relativo sito.

Per quanto l'articolo sia redatto con la massima cura, nè «Linux&C.» nè l'autore potranno ritenersi responsabili degli usi e/o abusi delle informazioni riportate.

Programma: crond

Sistemi vulnerabili: RedHat, Caldera, Debian.
Data: 27 Agosto 1999
Impatto: un utente locale può guadagnare i privilegi di root crond ha un "buffer over flow" che può essere "exploitato" creando un crontab con la variabile di ambiente MAILTO formattata in maniera particolare. Dato che il crond gira con i permessi di root è teoricamente possibile eseguire operazioni arbitrarie con i permessi di root.
Soluzione: aggiornare il proprio crond all'ultima versione.

Programma: irc epic 4

Versione: >= *pre1.034 e < pre2.003-19990718*
Sistemi vulnerabili: tutti.
Data: 27 Agosto 1999
Impatto: denial of service possibile.
L'autore di epic4 ha riportato una vulnerabilità di epic4 per tutte le versioni sopra indicate. È possibile far crashare il client irc e forse fargli visualizzare dei caratteri arbitrariamente scelti da chi effettua l'attacco. È interessante notare come questo sia uno dei pochissimi problemi di sicurezza notati su irc epic da molto tempo a questa parte.
Soluzione: aggiornare il proprio client irc epic all'ultima versione.

Programma: trn

Sistemi vulnerabili: tutti tranne OpenBSD (non segue i link simbolici sotto /tmp).
Data: 20 Agosto 1999
Impatto: È possibile riuscire a sovrascrivere qualunque file posseduto dall'utente che esegue trn. Trn utilizza un nome di file temporaneo "hardcoded", ovvero scritto direttamente dentro i sorgenti del programma. Un utente "cattivo" puo' creare un link simbolico con lo stesso nome dentro /tmp che punta al file che si vuole far sovrascrivere.
Soluzione: aggiornare trn all'ultima versione disponibile.

Programma: libtermcap

Sistemi vulnerabili: tutti i sistemi che utilizzano la libtermcap *(Debian non la utilizza di default, ma è possibile aver compilato un programma utilizzandola o scaricato un binario che la utilizza).*
Data: 18 Agosto 1999
Impatto: È possibile ottenere i privilegi di root localmente. La libtermcap contiene un buffer over flow. Questo significa che utilizzando una entry termcap particolarmente formattata si può riuscire a far eseguire codice arbitrario ai programmi che utilizzano questa libreria. In casi molto particolari tale problema potrebbe compromettere anche la sicurezza remota del sistema perchè alcuni software vulnerabili potrebbero accettare TERM settato a ../../../tmp/xxx. **L'exploit locale per xterm è gia' stato pubblicato, quindi attenzione!**
Soluzione: aggiornare la libtermcap all'ultima versione. Nel caso della Debian se siete certi di non utilizzare mai programmi basati su tale libreria potete evitare di patchare.

Programma: telnetd & libncurses

Versione: *telnetd < 0.12-3 & libncurses < 4.2-3*
Sistemi vulnerabili: Linux per telnetd, Linux ed altri per libncurses.
Data: 23 Agosto 1999
Impatto: È possibile costringere il telnetd a leggere qualunque file locale passandogli un "terminal type" utilizzando la celeberrima tecnica del ../../../, su alcuni sistemi forzando la lettura di alcuni file sotto il file system /proc si può riuscire a bloccare il sistema. **Sia chiaro che non c'è la possibilita' diretta di visualizzare il contenuto di questi file.**
Soluzione: fare l'upgrade del telnetd e delle libncurses.

Programma: Kdm

Sistemi vulnerabili: quelli che utilizzano Kdm con le impostazioni di default.
Data: 23 Agosto 1999
Impatto: Un attaccante remoto può visualizzare una schermata di login del sistema attaccato, ottenere una lista degli utenti del sistema e scavalcare i meccanismi di controllo degli accessi tipo tcpwrapper o /etc/securetty.
Soluzione: editare il file /etc/X11/kdm/Xaccess sostituendo le seguenti righe:

```
*                           #any host can get a login window
*    CHOOSER BROADCAST     #any indirect host can get a chooser
```

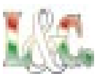
con le seguenti

```
#   *                           #any host can get a login window
#   *    CHOOSER BROADCAST     #any indirect host can get a chooser
```

ovvero è necessario semplicemente "commentare" le due righe che permettono la connessione da parte di qualunque host remoto.

Programma: cfingerd (tutte le versioni inferiori alla 1.4)

Sistemi vulnerabili: tutti.
Data: 14 Agosto 1999
Impatto: gli utenti locali possono eseguire programmi con i permessi di root. Cfingerd, se configurato con l'opzione ALLOW_EXECUTION (di default di solito è disabilitata), permette ad un utente normale di eseguire qualunque comando con i privilegi di root. C'è anche un buffer over flow probabilmente non exploitabile (per controllare se un host è vulnerabile prova "finger [più di 80 caratteri]@victim.org").
Soluzione: aggiornare a cfingerd 1.4, considerevolmente più sicuro dei suoi predecessori.



Salvatore Sanfilippo è esperto di sicurezza delle reti, si occupa di programmazione distribuita con il protocollo Tcp/Ip, sviluppo di software per il controllo e la sicurezza. Attualmente collabora con Alicom Italia (www.alicom.com). È raggiungibile all'indirizzo email s.sandiloppo@oltrelinux.com; la sua home page è http://www.kyuzz.org/antirez



open the WINDOW!

OPEN THE

LORENZO LAZZERI
l.lazzeri@oltrelinux.com

Programmatore C/C++, si occupa di amministrazione di sistema presso un ISP. Sviluppa applicazioni Web con PHP e SQL server.

Configurare il server X, ovvero la "veste" grafica del nostro Linux. Semplice ma non troppo, in particolar modo per un nuovo utente. Molto spesso quello che spaventa è l'enorme numero di domande che vengono poste... la risposta delle quali, alle volte, è davvero banale. Una guida per ridurre al minimo eventuali problemi.

KDE o Gnome? WindowMaker o Enlightment? Quante volte ci è capitato di assistere a un dibattito su quale sia il miglior window manager o il più facile ambiente? Quante volte ci siamo trovati a invidiare gli utilizzatori di altri sistemi operativi per la loro interfaccia grafica accattivante e facile da configurare (quando addirittura non è parte integrante dell'ambiente)? **X Window** (detto semplicemente X per gli amici) è il nome del sistema in grado di fornirci un piacevole ambiente a finestre sotto Linux. Può fare anche molte altre cose, ben più potenti e utili del semplice visualizzare le nostre applicazioni preferite incorniciate in accattivanti finestre. In questo articolo parleremo soprattutto di come fare a configurare X, nella sua incarnazione freeware, **Xfree86**.

> Configuriamo **X**, il **server** grafico di Linux, nella sua incarnazione freeware, **Xfree86**

Come funziona?

Una breve introduzione su cosa effettivamente è X e come funziona. **X è sostanzialmente un sistema client/server per la comunicazione di primitive grafiche**. Detto così non è semplice arrivare a capire come possa essere utilizzato per la visualizzazione di un ambiente grafico, ma è proprio in questa sua caratteristica che risiede la sua **potenza** e **versatilità**.
Il server è un'applicazione che **comunica** con **l'hardware** grafico (la scheda grafica, per intenderci) e sa come fargli visualizzare quello che vuole. Il **client** è qualsiasi **applicazione** scritta per funzionare sotto X Window
A grandi linee, quando viene lanciata un'applicazione (per esempio Netscape), questa si collega al server X e gli **comunica cosa** deve far comparire sullo schermo. Tutta la comunicazione avviene attraverso un **socket**, quindi è possibile anche una situazione nella quale l'applicazione risiede su una macchina anche fisicamente distante (ma connessa in rete!) dal server e addirittura visualizzare sullo stesso server **applicazioni lanciate** su più computer differenti. In ogni caso nella maggior parte dei casi, e soprattutto parlando di PC casalinghi, il server e il client (X e le applicazioni tipo Netscape) saranno in esecuzione sulla stessa macchina; questo porta spesso a definire X Window come l'interfaccia grafica di Linux, quando in realtà è la definizione di un sistema di comunicazione per primitive grafiche (il protocollo si chiama **X11**, nelle varie Release).

Le informazioni necessarie...

La maggior parte delle distribuzioni di Linux viene in aiuto dell'utente nella fase di configurazione di X (anche se probabilmente parlare di "aiuto" è un po' troppo ottimista)
Lle domande che vengono poste possono sembrare **strane** e/o **assurde**: viene, ad esempio, chiesta la frequenza di **refresh** del nostro monitor.
Analizzandole nel dettaglio vedremo che **non** è un compito così difficile capire le domande e sapere quindi cosa rispondere. Innanzitutto è consigliato avere a portata di mano il **manuale** del monitor; se purtroppo il nostro video è vecchiotto e quindi talel manuale si è incastrato chissà dove in casa, è comunque possibile reperire le stesse informazioni necessarie dal sito Internet del produttore. E' necessario anche conoscere quale **tipo di scheda video** abbiamo nonché il suo **modello**. Nel caso estremo di non disponibilità delle informazioni, sarà comunque possibile configurare X Window in modo **prudente**.



Figura 1: Il sito www.Xfree86.org, dove reperire i necessari aggiornamenti riguardo alle nuove schede non ancora supportate.

Si, ma... da dove si configura?

Innanzitutto è necessario aprire una sessione sul proprio sistema come utente **root**. Il programma da lanciare si chiama **xf86config**:

```
# /usr/X11/bin/xf86config
```

E' importante scrivere il **percorso di xf86config per intero**, visto che non è detto che la directory dove risiede sia nel nostro path di ricerca.

Le domande...

xf86config presenta subito la prima domanda, alla quale non c'è che da rispondere premendo il tasto Invio sulla tastiera. La prima vera domanda che ci verrà posta, alla quale è consigliato di **non** sbagliare la risposta, è quale tipo di mouse protocol vogliamo utilizzare. In soldoni, vuol solo sapere che tipo di mouse abbiamo: se con porta PS/2, seriale, con BUS (scheda dedicata dentro al PC) Microsoft oppure Logitech, eccetera. Se si possiede un mouse con porta di tipo seriale, rispondere 1 o 2 a seconda del numero di tasti disponibili. Nel caso di mouse BUS la risposta corretta è la 3, nel caso di mouse PS/2 è la 4. **Non** fatevi ingannare nel caso di un mouse (seriale o PS/2) della **Logitech**: a meno che il vostro puntatore non sia veramente vecchio, dovete fare affidamento al tipo di porta utilizzata e **ignorate** le opzioni specifiche dei mouse Logitech.
La domanda successiva chiede se vogliamo che il terzo bottone, nel caso il nostro mouse ne abbia solamente due, sia emulato dalla pressione contemporanea dei due presenti. Rispondere Y solo se avete due pulsanti.
A questo punto viene richiesto di specificare tramite quale device è possibile accedere al mouse. Solitamente esiste nella directory **/dev/** un device (in realtà è un link) chiamato **mouse**, nel caso specificare quindi **/dev/mouse**. Altrimenti, se il mouse è seriale o PS/2, considerate la seguente tabella e rispondete di conseguenza:

/dev/ttyS0	COM1
/dev/ttyS1	COM2
/dev/ttyS2	COM3
/dev/ttyS3	COM4
/dev/psaux	Porta PS/2

Mouse pronto... adesso la **tastiera**!
Eh sì, tocca anche dire che tipo di tastiera si possiede. La prima domanda serve a specificare se ci si vuole avvalere di alcune estensioni di X Window per la gestione della tastiera: è vivamente consigliato rispondere **sì** (Y), soprattutto se si ha una tastiera italiana.
La domanda successiva è una di quelle facili: basta premere invio!
Di seguito, invece, viene chiesto il tipo di tastiera usata. Per le tastiere con layout italiano, consiglio di rispondere **10** alla prima domanda, **1** o **2** a quella successiva (sul numero di tasti), e **9** all'ultima che chiede il tipo di mappatura (l'opzione n. 9 è quella relativa alla nostra lingua).

> «I primi dispositivi da configurare sono il mouse e la tastiera.
> Per il mouse, dovete fare affidamento al tipo di porta utilizzata»

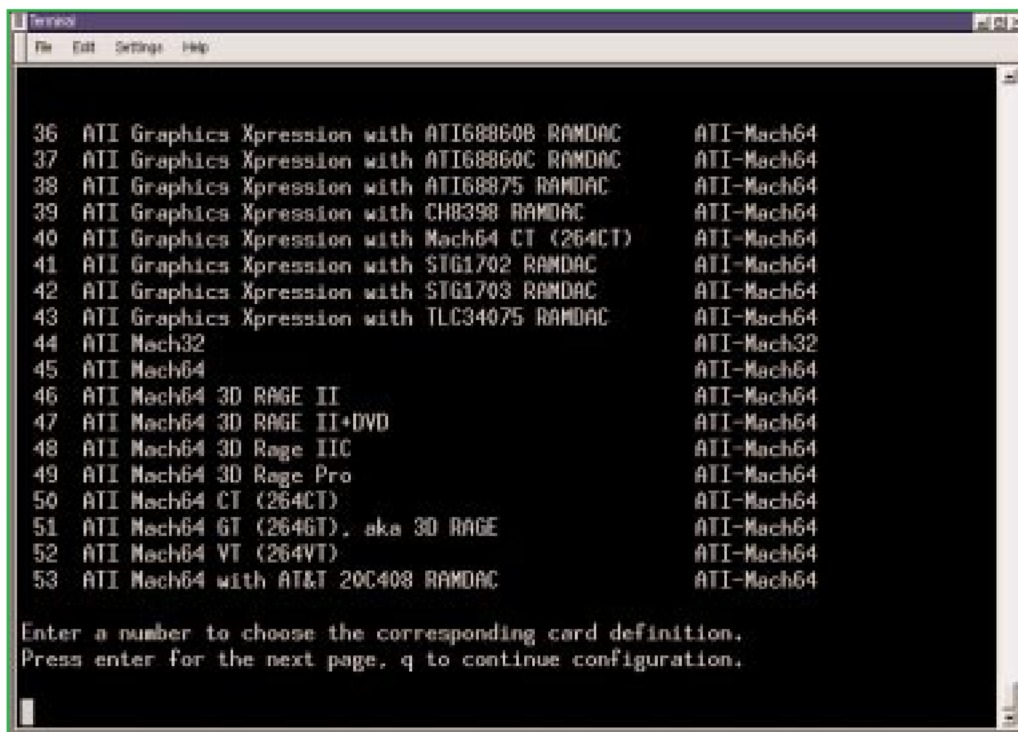


Figura 1: La scelta del modello di scheda video avviene selezionando il numero corrispondente da un elenco costantemente aggiornato.

Il Monitor

Xf86config vuole a questo punto sapere che tipo di **monitor** andremo a usare, così da poter scegliere la velocità di **refresh** dello schermo più adatta.
E' necessario avere a portata di mano il manuale del nostro monitor, magari aperto alla pagina contenente la tabella con le frequenze verticali supportate alle diverse risoluzioni. Si dovrà quindi scegliere il tipo di monitor che più si avvicina alle caratteristiche indicate a proposito del nostro, o al limite specificarle manualmente se nessuna delle scelte proposte sono perfettamente combacianti (scelta vivamente sconsigliata in quanto è possibile danneggiare **fisicamente** il monitor!).
In generale, solitamente è sufficiente controllare quale frequenza massima il nostro monitor è in grado di supportare e scegliere l'opzione più vicina (consiglio: meglio scegliere un'opzione che indichi una frequenza leggermente **inferiore** piuttosto che una anche solo leggermente **superiore**).
Nel caso comune di un monitor SuperVGA che arrivi a una risoluzione di **1024x768** a **60 Hertz** (non interlacciato), l'opzione giusta è la n.**6**, nel caso di un monitor un po' più vecchio che supporta sì i 1024x768 ma in modalità **interlacciata** (la schermata viene disegnata in due passate successive), l'opzione da scegliere è la **4**.
Infine, se siete fortunati da avere un monitor capace di risoluzioni anche di 1280x1024, scegliete quella con la frequenza di aggiornamento indicata nel manuale.
Ripeto: scegliere una frequenza superiore a quella massima supportata dal video per quella risoluzione lo può danneggiare anche in modo serio.
Scelta l'opzione giusta è come sempre necessario premere il tasto Invio.
E' finita? Non proprio... xf86config adesso chiede il range di frequenze per la sincronia **orizzontale**. Quelle proposte sono le più comuni, anche se è pos-



sibile specificare un range che il nostro monitor supporta ma non viene elencato.
In ogni caso, tale informazione è riportata sul manuale proprio con la medesima dicitura, probabilmente nella paginetta riassuntiva delle caratteristiche tecniche.
Se tale informazione non è disponibile, potete provare a scegliere l'opzione n. **2**, buona anche per i vecchi monitor capaci di risoluzioni di 640x480, 800x600 e 1024x768 (interlacciato). Anche in questo caso valgono comunque gli avvertimenti riportati più su a proposito della possibilità di danneggiare il monitor.
A questo punto un po' di relax, se vogliamo: le tre domande successive serviranno a dare un nome al monitor. Sarà quindi chiesto **un'identificativo**, il **nome del produttore** e il **modello del monitor**. Non sono informazioni essenziali e comunque non vengono controllate: potete lasciare anche in bianco premendo Invio a tutte le tre domande.

> «XFree86 viene supportato da volontari e distribuito gratuitamente, è quindi improbabile che la scheda 3d ultimo modello sia supportata!»

La scheda grafica

Il quiz da completare per la configurazione di X Window passa adesso alla fase di riconoscimento (si fa per dire...) della **scheda grafica**.
Per fortuna xf86config ci chiede se vogliamo scorrere **l'intero database** delle schede grariche supportate alla ricerca della nostra. Visto che Xfree86, come il nome stesso indica, viene sviluppato da **volontari** e **distribuito gratuitamente**, è improbabile che la scheda grafica acceleratrice 3d ultimo modello sia tra quelle supportate. E' quindi consigliato, prima di procedere all'acquisto di una scheda grafica, controllare che sia supportata da Xfree86 se non si vuole che con Linux venga utilizzata come una vecchia VGA.
A meno che non si conosca già il numero della scheda grafica (nel caso non sia la prima configurazione di X), rispondete **Y** alla domanda, così da farsi elencare, pagina per pagina, le schede grafiche.
Come consiglia xf86config stesso, se **non** si trova la nostra scheda grafica, basta scegliere dal database una scheda che sia lo stesso **modello** della nostra piuttosto che una che abbia un **nome simile**, ma non **uguale**, poiché potrebbe in realtà differire di molto a livello di hardware.
Premendo Invio si scorre ciclicamente tutto il database. Quando troverete la vostra scheda potete digitarne il numero identificativo e premere Invio.

Xf86config chiederà a questo punto quale server X utilizzare, si rende necessaria una precisazione. Xfree86, a differenza di altri ambienti e/o sistemi operativi, **non consente** di aggiungere il driver della scheda video come modulo a parte; questo perché i **driver** vengono **codificati** direttamente all'interno del **server**. Nella distribuzione standard vengono forniti perciò **diversi server** (una decina), ognuno dei quali col supporto per una o più famiglie di schede video. E' importante perciò che sul proprio sistema sia **installato** quello adatto.
Quando xf86config chiede quale server usare, quindi, è consigliato rispondere **5**, cioè Choose the server from the card definition, **XF86_XXXXX**.
Dopodiché, il tool di configurazione richiederà se si vuole creare un link nella directory proposta per quello specifico server: questo si rende necessario perché gli applicativi si aspettano che il server si chiami semplicemente X, mentre il server di solito ha un nome tipo XF86_SVGA. Rispondere **Y** a questa domanda, e anche a quella successiva relativa a dove si vuole creare quel link.
Abbiamo quasi finito di configurare la scheda grafica: ci verrà chiesto adesso di specificare di quanta memoria video la nostra scheda è dotata. Se tra le opzioni preimpostate non c'è la quantità esatta, specificarla scegliendo l'opzione **6**, ricordandosi che deve essere indicata una quantità che sia una potenza di **2** (ad esempio, per **16** Mb è necessario specificare **16384**).
Come è successo per il monitor, anche per la scheda video è possibile definire **nome**, **marca** e **modello**: premendo semplicemente Invio alle domande successive si accetteranno le definizioni proposte.

Ma che domande sono?

Le ultime domande sono le più **difficili**, e nella maggior parte dei casi anche le più **inutili**. Vediamole nel dettaglio.
Xf86config ci chiede se la nostra scheda grafica è dotata di un **Clock chip programmabile**, ma ci suggerisce che la maggior parte delle schede non ne possiede uno. In tal caso premere **Invio** senza troppi sensi di colpa. Se invece siamo a conoscenza di questa particolare dotazione della nostra scheda video, e il Clock chip è elencato nella lista, specificarlo.

> «Le ultime domande sono le più difficili e, nella maggior parte dei casi, anche le più inutili.»

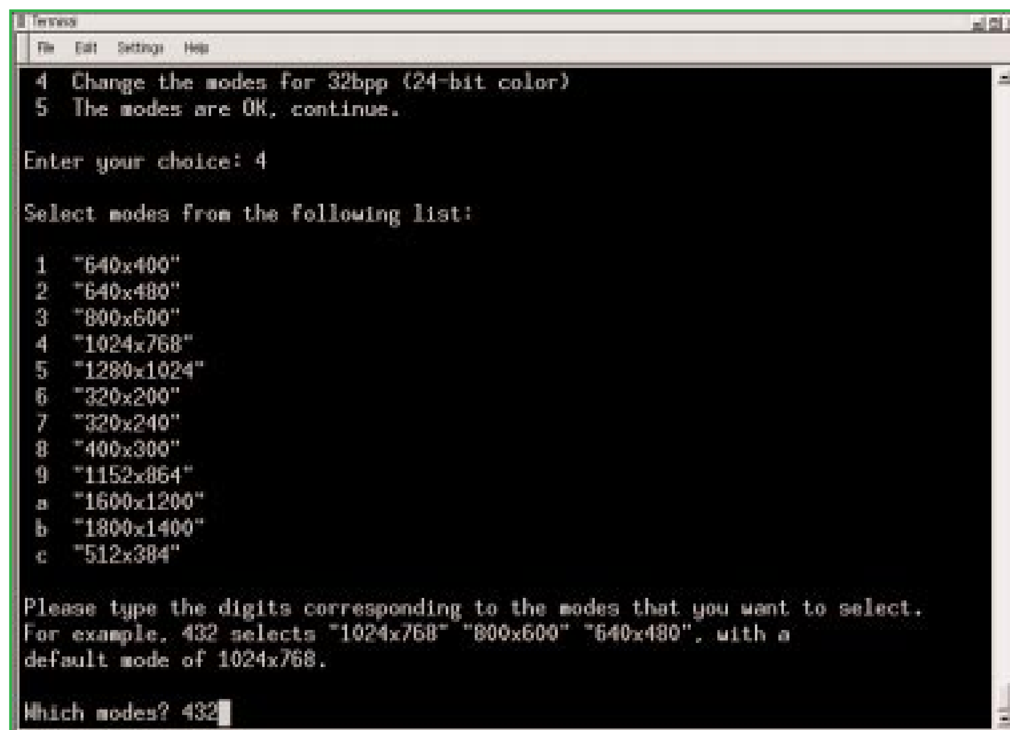


Figura 2: La selezione delle risoluzioni di visualizzazione, indicando come primo numero (4 nell'esempio) la modalità di default.

In funzione della scheda grafica scelta, può capitare che xf86config faccia anche un'altra domanda dall'aspetto difficile e dalla risposta facile: quale **RAMDAC** abbiamo? Se lo conosciamo (ma **solo** se lo conosciamo) ed è presente nell'elenco, specificarlo, altrimenti premere **q** e proseguire. Con la domanda successiva xf86config intende chiederci se vogliamo





Figura 3 e 4: Alcuni immagini «grabbate» dal desktop. Niente male, che dite? Adesso sono gli altri s.o. che devono venire a prendere ripetizioni dal nostro Linux in fatto di grafica!

consentirgli di tentare varie **temporizzazione** per i modi grafici. Nella maggior parte dei casi, con hardware **sufficientemente moderno**, questo **non** è necessario e anzi, **sconsigliato**. Rispondete quindi N.

E' possibile a questo punto scegliere i modi grafici (le risoluzioni e le profondità di colore) preferiti: per farlo, selezionare l'opzione giusta (per cambiare l'ordine dei modi grafici per 16 bit colore, per esempio, selezionare 2), e di seguito elencare uno dopo l'altro i modi; per esempio per ottenere i modi in questo ordine, "640x480" "1024x768", digitare 24. Di seguito verrà chiesto se si vuole che il desktop sia più grande della risoluzione scelta, facendo in modo che il monitor sia una sorta di "finestra sul desktop"; se si desidera usufruire di questa possibilità rispondere di sì (Y).

Si possono specificare modi grafici per ogni profondità di colore, fino a che non si preme Invio per proseguire.

> «L'ultima domanda chiede la conferma affinchè xf86config crei per noi il file di configurazione nella directory /etc»

Abbiamo finito!

Eccoci giunti all'ultima domanda, quella fatidica: vogliamo che xf86config **crei** per noi il file **XF86Config** nella directory /etc ? Questo significa chiaramente che fino ad adesso niente è stato scritto e che quindi si può tranquillamente rispondere di no e ricominciare da capo. Nel caso in cui invece siamo più o meno sicuri delle risposte date, possiamo consentire a xf86config di scrivere il vero file di configurazione di Xfree86.

Si ma... e adesso?

Abbiamo il nostro Xfree86 bello configurato, possiamo pure usarlo. Come? Per lanciare X si deve usare il comando **startx** (non importa essere root):

```
$ startx
```

Se si vuole usare una **profondità di colore** specifica (altrimenti parte di default con 256 colori) si può usare l'opzione -bpp:

```
$ startx -- -bpp 24
```

Se alla partenza X si blocca su uno schermo grigio (in realtà son puntini neri e bianchi...) con un cursore a forma di X senza dare segni apparenti di vita, provate a muovere il mouse e/o clickare: può essere che il **window manager** (in parole povere il gestore delle finestre) attualmente in uso sia molto scarno (di default è TWM): in questo caso dovrebbe apparire un menu. Se invece siete nel panico perché non succede assolutamente nulla, potete usare la combinazione di tasti **CTRL+ALT+BackSpace** per chiudere X.

Il window manager

Nelle più recenti distribuzioni viene installato di default un ambiente operativo, Gnome o KDE, già **preconfigurato** che aspetta solo che parta X per mostrarsi in tutta la sua bellezza e semplicità di uso. Quindi, una volta aperto X dovrebbe comparire un'interfaccia grafica abbastanza intuitiva. Nel caso invece che questo non accada, sarà necessario scegliersi un window manager e/o un ambiente operativo, installarlo e configurarlo. Tutto ciò esula dallo scopo di questo articolo, ma sarà ampiamente discusso nei prossimi numeri.

Nei prossimi numeri...

Sempre nel campo della grafica, vedremo come utilizzare il frame buffer ed accedere così alla memoria video della scheda grafica tramite un device.



PROBLEMI DI CONVIVENZA

Quando vogliamo installare Linux ma non sappiamo DOVE!

l.lazzeri@oltrelinux.com

Ci sono sostanzialmente **due** tecniche per installare Linux: *da solo, o in compagnia*. L'installazione di Linux come **secondo sistema operativo** si rende necessaria quando non si può o non si vuole **rinunciare** a quello che abbiamo usato fino ad adesso, vuoi per motivi professionali, vuoi per motivi di "compatibilità" per lo scambio di dati, vuoi per motivi puramente ludici, anche se stanno uscendo davvero ottimi prodotti in questo senso :-).

Linux da solo

Nel caso si decida che Linux sia l'unico sistema operativo **degno** di girare sulla nostra macchina, la fase di preparazione del disco rigido è piuttosto semplice. Si tratta di creare **due partizioni**: una che andrà a contenere il **sistema operativo vero e proprio**, i programmi, i dati; l'altra che sarà utilizzata come spazio di **swap** (*memoria virtuale*).
Qualsiasi sia la **distribuzione** che stiamo installando, vedremo come i passi da seguire saranno gli **stessi**: quello che cambia è il programma da utilizzare e la sua semplicità di uso.
Prima di procedere all'installazione, quindi, si dovranno creare due partizioni, **cancellando** quelle esistenti prima di procedere alla suddivisione.
L'hard disk con Linux viene chiamato /dev/hda (nel caso di dischi IDE) o /dev/sda (nel caso di dischi SCSI). L'ultima lettera **varia** al variare della posizione che il disco ha nella **catena IDE o SCSI**. Vediamo a tal proposito la tabella 1.
La creazione delle partizioni avviene con programmi tipo **fdisk** o **cfdisk** (tipicamente con le distribuzioni Slackware e Debian, anche se tali tool sono presenti pure sulle altre), con **DiskDruid** (sulle RedHat, tra l'altro molto semplice da usare).
Si dovrà fare attenzione soprattutto alla quantità di spazio da **assegnare** alla partizione di swap: di solito si consiglia di crearla grande il **doppio** della quantità di **memoria RAM disponibile**; ad esempio se si hanno 32 Mb di RAM la partizione di swap è bene che sia di 64Mb. Non ha comunque molto senso eccedere i 128Mb di swap: merita piuttosto espandere la memoria on-board.
Benché sia bene creare la partizione di swap **in fondo al disco**, e perciò per **ultima**, è necessario decidere **prima** quanto dovrà essere grande in modo da **non** utilizzarne lo spazio per quella di **root** (la prima).
Procedere quindi creando la prima partizione, che dovrà essere di tipo **"Linux native"**. Tale partizione verrà chiamata con un nome tipo /dev/hda1 o /dev/sda1 a seconda del tipo di controller del disco (/dev/hdb1 o /dev/sdb1 se è il secondo disco, e così via).
Infine creare la partizione di swap, a cui assegnare il tipo **"Linux swap"**: è probabile che il tipo di partizione in questo caso vada specificato in quanto di **default** i tool la creano "Linux native" (è comunque presente una legenda con i codici e le specificazioni del tipo di partizione, quindi non preoccupatevi troppo).
Durante la fase di **creazione** delle partizioni, ci verrà chiesto se vogliamo che queste siano **primarie** o **estese**. Linux può essere installato su entrambi i tipi di partizioni, anche se normalmente si utilizzano quelle **primarie** tranne necessità particolari: non è infatti possibile crearne più di **quattro**.
Una volta finito di fare a *fettine* il nostro disco, sarà necessario **salvare i cambiamenti apportati** (quindi, fino a questo punto, niente è stato modificato nella tabella delle partizioni... non vi disperate in caso di errori...) e **riavviare** la macchina, procedendo **all'installazione vera** e propria, che si occuperà anche di **formattare** le partizioni.

Tabella 1

Disco IDE	Disco SCSI	Nome in DOS
/dev/hda	/dev/sda	C:
/dev/hdb	/dev/sdb	D:

Corrispondenza tra nomi dei dischi rigidi sotto MsDos (o Windows) e Linux.

Linux con Windows: due modi differenti.

Se decidiamo che l'altro sistema operativo comunque ci è **necessario** e non vogliamo **esiliarlo** del tutto dalla nostra macchina, dobbiamo istruire Linux a conviverci, **preparandolo**. Windows, invece, neanche se ne **accorgerà**.
Ci sono due modi di far convivere Linux con Windows, uno **meno** performante ma più adatto a un test indolore, l'altro più **invasivo** ma dal risultato simile ad avere Linux quale singolo inquilino del nostro disco rigido.
E' infatti possibile **ridimensionare** la partizione presente (*con qualche condizione da soddisfare*) e creare quelle su cui installare Linux (la sua più quella di swap), oppure installarlo direttamente sul filesystem di Windows (o del DOS).

Aggiungi un posto alla tavola... delle partizioni!

Nel caso scegliessimo di ridimensionare la partizione, dobbiamo fornirci di un programma in grado di farlo: **FIPS.EXE**, free, di cui sono disponibili i sorgenti e che funziona egregiamente; oppure un tool commerciale tipo Partition Magic.
Nel caso di FIPS, che prenderemo in considerazione per via della sua **onnipresenza** su praticamente **tutte** le distribuzioni, prima di poterlo utilizzare è necessario **deframmentare** il disco rigido così da raggruppare tutto lo **spazio disponibile** in fondo al disco.
Tale operazione può essere effettuata anche con l'utility in dotazione a Windows 95/98 **"Utilità di deframmentazione dei dischi"** *(in Windows NT non è invece presente alcun tool di default: si può ricorrere a Diskepeer oppure Norton Speed Disk)*, disattivando la **memoria virtuale** (vivissimamente consigliato! le fasi da seguire sono *Start -> ControlPanel -> System -> Performance ->*



VirtualMemory -> Attivare il settaggio manuale della memoria -> Selezionare il check Disable virtual memory -> riavviare e deframmentare il disco).
Purtroppo la riuscita **non è sempre assicurata**, in quanto durante la sua esecuzione, Windows **non** consente di spostare alcuni file, che se si dovessero trovare in prossimità della fine del disco comprometterebbero il risultato dell'operazione.
Una volta deframmentato lo spazio disponibile si potrà procedere con FIPS (che è un programma da eseguire **in modalità DOS** e **non** dentro una **finestra** DOS).
FIPS innanzitutto esegue **l'analisi** del disco rigido e poi chiede se vogliamo fare una copia di backup su **dischetto** della tabella delle partizioni: **è vivamente consigliato farla, se qualcosa dovesse andare male saremmo in grado di ripristinare tutto come prima.**
Dopo la copia di backup, FIPS, con l'aiuto dei tasti freccia, ci consente di decidere quanto dello spazio disponibile **rilasciare** (sarà infatti reso come spazio non partizionato, in pratica rimpicciolendo la partizione esistente): è necessario non scordasi che se si sceglie di ripartizionare **tutto** lo spazio disponibile non rimarrà neanche un byte a Windows. Dosare **bene** i Megabyte, quindi!
Una volta completato, riavviare la macchina e procedere con un tool tipo fdisk o cfdisk o DiskDruid al partizionamento.



Figura 1: cfdisk, uno dei tool per partizionare l'hard disk, presenta un disco con numerose partizioni, tra cui una Windows 95, una Windows NT, e diverse Linux.

E' ovvio che una partizione risulterà già creata: /dev/hda1 o /dev/sda1 a seconda del **controller** del disco. Fare attenzione a non cancellare questa partizione: cancelleremmo **completamente** Windows e tutti i dati presenti sul suo spazio (*in realtà si può tentare di rimediare, vedremo in futuro come*).
Dovremo quindi creare due partizioni: quella per Linux e quella di swap. Seguire le stesse regole esposte per il caso di Linux come unico sistema operativo installato, nel paragrafo precedente, facendo attenzione che le due ulteriori partizioni si chiameranno /dev/hda2 e /dev/hda3 (o gli analoghi per SCSI).

Linux ospite?

Personalmente penso che **non se lo meriti**, soprattutto per via delle **performance nettamente inferiori** rispetto al metodo precedente, ma può capitare la necessità di installare Linux su una partizione già esistente, formattata con il filesystem di Windows (FAT o FAT32): ottima, questa soluzione, se si vuole **provare questo sistema** del quale tutti parlano. La **disinstallazione**, infatti, in questo caso, davvero banale (si tratta semplicemente di cancellare una directory, cosa che si può fare da dos o da windows).
Linux si installerà quindi in una directory (C:\LINUX) che al momento del **boot** vedrà come root del suo filesystem.
Benché il filesystem vero e proprio sia FAT (o FAT32), **quindi con le limitazioni di un sistema monoutente**, Linux **emulerà** le funzioni necessarie all'uso in **multiutenza** (ownership e permessi in primis). Resta inteso che quando sarà in esecuzione Windows le directory e i file di Linux saranno **leggibili e scrivibili a prescindere dai permessi assegnati**, con tutti gli **effetti** che ne derivano: sarà possibile quindi cancellare completamente la directory, leggerne il contenuto, modificarlo (anche se il risultato non è detto sia qualcosa di utilizzabile), con il rischio di perdita dei dati.
Inoltre, ne **risente** la velocità di accesso ai file, in modo tanto più sensibile quanto più si farà uso di applicazioni che leggono e scrivono su disco.
Rimane comunque un buon metodo per chi vuole provare Linux senza **stravolgere** le partizioni del proprio hard disk, nonché per condividere lo spazio disponibile tra i due sistemi operativi; cosa che con l'uso delle partizioni non è possibile (almeno, non del tutto). In ogni caso, per installare Linux con questo metodo **non** si deve preparare in alcun modo il disco rigido, anche se una deframmentazione è comunque consigliata.
L'unico accorgimento è di istruire il programma di installazione della propria distribuzione per utilizzare il filesystem chiamato **UMSDOS**.

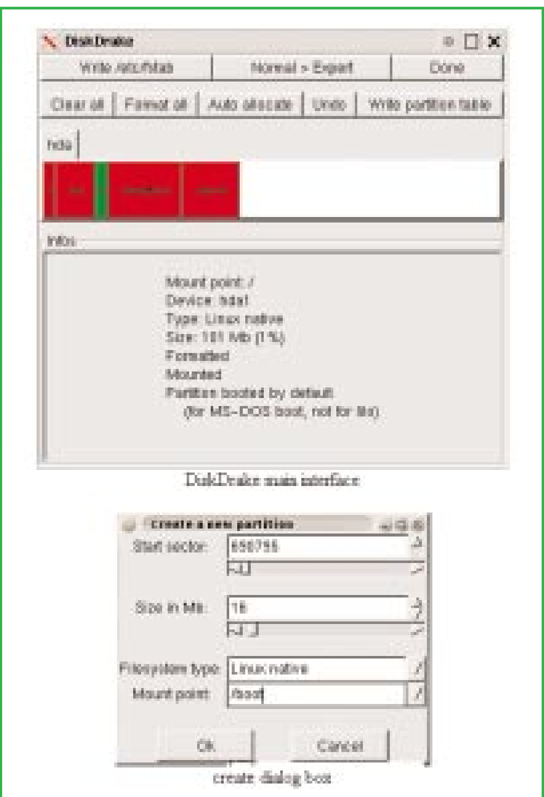


Figura 2: DiskDrake, il tool grafico di ripartizionamento dei dischi distribuito insieme a Linux-Mandrake.



Masterizzare fa-cì-le

FELICE MAINOLFI
f.mainolfi@oltrelinux.com

25 anni, laureando in Ingegneria Elettronica, si occupa di sistemi distribuiti e sicurezza in ambiente unix.

*A torto linux e' ritenuto la "**cenerentola**" della masterizzazione, forse a causa del numero esiguo di programmi per il "**burning**" dei cd: in questo articolo cerhiamo di sviscerare i segreti delle varie utility che abbiamo a disposizione per una corretta masterizzazione.*

Il continuo **abbassamento** dei costi dei supporti e dei **masterizzatori** ha reso particolarmente **conveniente** la scelta di effettuare il **backup** dei dati su **cd-rom**. La possibilità di fare in casa il proprio cd ha però provocato l'esplosione del già fiorente mercato della **pirateria**. In questo articolo supporremo che le **copie** che effettuerete siano solo da prodotti originali e al solo scopo di possedere una copia di **backup** di prodotti **regolarmente** acquistati.

Standard di mercato

La storia dell'evoluzione dei cd-rom passa attraverso una serie di standard che ne descrivono sia il formato fisico sia il file system usato. Di seguito vengono riassunti quelli di nostro interesse:

Red Book: contiene le specifiche per il formato fisico dei cd audio.
Yellow Book: descrive lo standard per la memorizzazione di dati su cd.
Orange Book: è la sintesi dei due precedenti per la memorizzazione su CD-R (cd registrabili, quelli che usiamo normalmente).
ISO9660: specifica il file system da usare su un cd-rom ed èindipendente dalla piattaforma.
Rock Ridge Extensions: sono una serie di estensioni all'ISO9660 per l'utilizzo in ambito UNIX.
Joliet Extensions: le estensioni di Microsoft a ISO9660.
El Torrito Extensions: consentono la creazione di cd dai quali sia possibile il boot.

Il file system ISO9660 è "**MSDOS-oriented**" ed in quanto tale risente di tutte le **limitazioni** tipiche dell'msdos (mancanza dei **permessi** di accesso, limitazioni sulla **lunghezza** dei nomi dei file, **profondità** delle directory limitata...). Per far fronte a questo problema sono nate le estensioni Rock Ridge che incapsulano nell'ISO9660 gli aggiornamenti necessari al corretto **backing-up** in **UNIX**. La stessa **Microsoft** ha esteso con le Joliet il file system originario per adeguarlo alle caratteristiche dei suoi Windows.
Per quanto ci riguarda possiamo masterizzare da linux in una qualsiasi di queste varianti.

Prima di cominciare...

...bisogna configurare correttamente il kernel (vedi riquadro 1).
Inoltre sono necessari i seguenti programmi:

Cdrecord: è il programma per la masterizzazione vera e propria.

Riquadro 1

UNO SGUARDO AL KERNEL

Come prima cosa, dopo aver montato fisicamente il masterizzatore e la sua interfaccia, bisogna compilare il kernel affinchè lo riconosca. Supponiamo di avere un masterizzatore con interfaccia SCSI: dobbiamo assicurarci in fase di configurazione del kernel che siano abilitate le seguenti voci nella sezione "SCSI support":

SCSI support
SCSI CD-ROM support
SCSI generic support

e controllare che nella sezione "SCSI low-level drivers" sia settato il driver del controller SCSI che si possiede.
Se, invece, il masterizzatore è fornito di un'interfaccia IDE allora è necessario che sia abilitata la voce:

SCSI emulation support

nella sezione "Block devices"; in questo modo la periferica IDE verrà pilotata alla stregua di una SCSI. A questo punto si può procedere alla ricompilazione del kernel e al rebooting. Potreste aver bisogno di generare i file /dev relativi ai dispositivi scsi generici; per farlo digitate:

```
cd /dev
MAKEDEV sg
```

Nell'articolo si suppone che usiate una versione del kernel uguale o successiva alla 2.0.34: versioni precedenti possono dare problemi nella gestione delle periferiche SCSI.



cdda2wav: estrae le tracce audio dal cd sorgente e le memorizza in file di tipo .wav
mkisofs: costruisce un'immagine ISO9660 dei file da trasferire su cd.

Il riquadro 2 fornisce le istruzioni per la corretta istallazione. L'ultimo passaggio è quello dell'identificazione del masterizzatore ed eventualmente di altri cd-rom sul bus SCSI; in altri termini è necessario conoscere la terna scsi-bus, id, lun associata al masterizzatore. Questo è possibile analizzando l'output del comando **dmesg**, cercando linee del tipo:

> «Nella masterizzazione in ambito Linux, è di fondamentale importanza conservare le informazioni relative ai permessi dei files»

```
Vendor: PLEXTOR
Model: CD-ROM PX-32TS
Rev: 1.02
Type: CD-ROM
ANSI SCSI revision: 02
Detected scsi CD-ROM sr0 at
scsi0, channel 0, id 1, lun 0

Vendor: TEAC
Model: CD-R55S
Rev: 1.0Q
Type: CD-ROM
ANSI SCSI revision: 02
Detected scsi CD-ROM sr1 at
scsi0, channel 0, id 3, lun 0
```

nel mio caso la prima entry individua il mio cd-rom scsi che è situato sul bus 0 con id 1 e lun 0, mentre la seconda individua il masterizzatore posto sul bus 0 con id 3 e lun 0. Non vi allarmate se il kernel riconosce anche il masterizzatore come cdrom: è del tutto normale.
Se avete un solo bus e siete sicuri che tutte le periferiche scsi hanno lun 0 potete ricavare l'id con il semplice comando

```
cdrecord -scanbus
```

che, inoltre, fornisce la stringa identificativa di ogni id.

Masterizzare cd-audio

La produzione di un cd audio richiede in primo luogo la preparazione delle tracce che verranno incise. Esse devono essere memorizzate in file .wav o .au e successivamente trasferite con cdrecord. Sebbene i file possano contenere brani di produzione personale, la maggior parte delle volte essi provengono da altri cd. Il trasferimento da cd a file .wav avviene tramite l'utility **cdda2wav**.
Il trasferimento completo delle tracce di un cd in file wav separati avviene mediante il comando:

```
cdda2wav -D0,1,0
         -t 1 -B -x
```

-D è seguita dal bus dall'id e dal lun del driver utilizzato per leggere le tracce audio (può essere anche il masterizzatore stesso).
-t 1 consente di trasferire tutte le tracce a partire dalla prima.
-B mette ogni traccia in un file .wav separato.
-x esegue il dump con la **massima qualità** (44.1 KHz, stereo, 16 bit).

La velocità di prelievo varia sensibilmente a seconda del costruttore del cd rom e/o masterizzatore.
Alla fine del processo avremo un insieme di file **audio_XX.wav** pari al numero di tracce del cd.
La registrazione vera e propria del cd viene effettuata dal comando:

```
cdrecord -v
 -dev=0,3,0 -speed=4
   -audio audio*.wav
```

-v consente di avere un rapporto dettagliato dell'avanzamento del processo di scrittura.
-dev=0,3,0 specifica il bus l'id e il lun del mio masterizzatore.
-speed=4 indica che la scrittura può avvenire alla velocità di 4x.
-audio audio*.wav istruisce cdrecord a prelevare i dati dai file audio_XX.wav e a inciderli come tracce audio.

Cdda2wav produce anche un file **.inf** per ogni **.wav** che contiene informazioni relative alla traccia corrispondente al .wav. Mediante il parametro **-useinfo** di cdrecord è possibile utilizzare queste informazioni per la masterizzazione.
Una volta iniziata la registrazione è **assolutamente necessario** evitare che si svuoti il buffer attraverso cui transitano i dati verso il masterizzatore. Per evitare errori di buffer underrun e il conseguente **danneggiamento irreversibile** del supporto è consigliabile mantenere al **minimo indispensabile** il carico di lavoro per i dispositivi di I/O durante la masterizzazione.
Naturalmente è possibile prelevare solo **alcune tracce** da **diversi cd** e masterizzarle **insieme** in modo da produrre una **compilation**.

Cd ISO9660

La masterizzazione dei dati in formato ISO9660 è possibile solo se prima si costruisce una immagine del cd in cui sono contenuti i file e le directory da inserire nella pubblicazione.
Mettiamoci prima nel caso, tipico dei **backup**, di avere una serie di file da trasferire sul cd. **Essendo in ambito linux è di fondamentale importanza**

Riquadro 2

COMPILARE CDRECORD

Il programma "principe" per masterizzare in linux è cdrecord. In questo articolo facciamo riferimento alla versione 1.8a21. Insieme al programma sono forniti anche mkisofs e cdda2wav necessari per la creazione delle immagini ISO e .wav delle tracce dei cd sorgenti. Prima di procedere alla compilazione probabilmente avrete bisogno di modificare il contenuto della directory DEFAULTS ed in particolare del file Defaults.linux; consiglio di reimpostare la seguente linea:

```
INS_BASE=    /opt/schily
```

in

```
INS_BASE=    /usr/local
```

La compilazione in se non comporta particolari problemi: basta un semplice "make install" affinchè il tutto venga configurato, compilato e installato.
Potete procurarvi l'ultima versione di cdrecord su internet in un qualunque archivio di programmi per linux, ad esempio http://flashnet.linuxberg.com

1

```
LFFoundry:~# cdrecord -v -dev=0,3,0 -speed=4 -isosize /dev/cdrom
Cdrecord release 1.8a21 Copyright (C) 1995-1999 Jörg Schilling
TOC Type: 1 = CD-ROM
scsidev: '0,3,0'
scsibus: 0 target: 3 lun: 0
atapi: 0
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : RELADR SYNC LINKED
Vendor_info    : 'TEAC    '
Identifikation : 'CD-R55S         '
Revision       : '1.0Q'
Device seems to be: Teac CD-R50S.
Using driver for Teac CD-R50S, Teac CD-R55S, JVC XR-W2010, Pinnacle RCD-5020 (t
ac_cdr50).
Driver flags   : SWABAUDIO
WRa Data 00 04 00 00
Buffer cap: 655360
Drive buf size : 655360 = 640 KB
FIFO size      : 4194304 = 4096 KB
Track 01: data   638 MB
Total size:      733 MB (72:38.44) = 326921 sectors
Lout start:      733 MB (72:40/71) = 326921 sectors
Current Secsize: 2048
Blocks total: 336225 Blocks current: 336225 Blocks remaining: 9304
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 1 seconds.
Waiting for reader process to fill input-buffer ... input-buffer ready.
Judging disk...done.
Calibrating laser...done.
Starting new track at sector: 0
Track 01: 638 of 638 MB written (fifo
Track 01: Total bytes read/written: 66
Writing  time: 1080.785s
Fixating...
Fixating time:   65.747s
cdrecord: fifo had 20433 puts and 2043
cdrecord: fifo was 0 times empty and 2
LFFoundry:~# 
```

2

```
#Cdda2wav version schily0.4_linux_2.2.8_i586_i586 real time sched. soundcard s
upport
AUDIOtrack pre-emphasis  copy-permitted tracktype channels
       1-19           no              no       audio     2
Table of Contents: total tracks:19, (total time 78:16.76)
  1.( 4:49.73),  2.( 4:08.07),  3.( 3:22.87),  4.( 3:36.29),  5.( 4:11.67)
  6.( 4:21.93),  7.( 3:49.00),  8.( 4:14.00),  9.( 4:55.00), 10.( 3:55.64)
 11.( 3:28.83), 12.( 4:01.07), 13.( 4:05.40), 14.( 4:06.93), 15.( 2:53.40)
 16.( 4:54.20), 17.( 4:07.87), 18.( 5:20.13), 19.( 3:52.73),
Table of Contents: starting sectors
  1.(       0),  2.(   21730),  3.(   40335),  4.(   55550),  5.(   71772)
  6.(   90647),  7.(  110292),  8.(  127467),  9.(  146517), 10.(  168642)
 11.(  186315), 12.(  201977), 13.(  220057), 14.(  238462), 15.(  256982)
 16.(  269987), 17.(  292052), 18.(  310642), 19.(  334652), lead-out(  352107
)
CDDB discid: 0x19125613
samplefiles size total will be 828156500 bytes. 19 audio tracks
recording 4694.76000 seconds stereo with 16 bits @ 44100.0 Hz ->'audio'...
overlap:min/max/cur, jitter, percent_done:
 1/ 1/ 1/       0   6%  track  1 successfully recorded
 1/ 1/ 1/       0  11%  track  2 successfully recorded
 1/ 1/ 1/       0  15%  track  3 successfully recorded
                                               ecorded
                                               ecorded
                                               ecorded
                                               ecorded
```

3

```
LFFoundry:/mnt/temp# dd if=/dev/cdrom of=immagine.cd
1304392+0 records in
1304392+0 records out
LFFoundry:/mnt/temp# cdrecord -dev=0,3,0 -speed=4 -data immagine.cd
Cdrecord release 1.8a21 Copyright (C) 1995-1999 Jörg Schilling
scsidev: '0,3,0'
scsibus: 0 target: 3 lun: 0
Device type    : Removable CD-ROM
Version        : 2
Response Format: 2
Capabilities   : RELADR SYNC LINKED
Vendor_info    : 'TEAC    '
Identifikation : 'CD-R55S         '
Revision       : '1.0Q'
Device seems to be: Teac CD-R50S.
Using driver for Teac CD-R50S, Teac CD-R55S, JVC XR-W2010, Pinnacle RCD-5020 (t
ac_cdr50).
Driver flags   : SWABAUDIO
Starting to write CD/DVD at speed 4 in write mode for single session.
Last chance to quit, starting real write in 1 seconds.
Track 01: Total bytes read/written: 667848704/667848704 (326098 sectors).
LFFoundry:/mnt/temp# 
```

Fig. 1: cdrecord - output durante il processo di copia *on-the-fly*.
Fig. 2: cdda2wav durante il ripping di un cd audio.
Fig. 3: copia di un cd attraverso una immagine ISO.

conservare le informazioni relative ai permessi dei file: per questo ed altri motivi è necessario produrre una immagine con le estensioni **Rock Ridge**. Mkisofs (MaKe ISO FileSystem) consente di costruire una tale immagine, pronta per essere trasferita al recorder.

I parametri di mkisofs sono molto **numerosi** e in questa sede parleremo solo dei più **importanti**.

Supponiamo di aver organizzato l'albero di directory da traferire al cd nella directory **/jobs/mycdrom** e di voler costruire l'immagine iso corrispondente; il comando da impartire sarà:

```
mkisofs -v -o immagine.cd -R
             -a -m core -V "MyVolume"
                         -P "LFF spa"
                 -p "Pinco  Pallino"
                         jobs/mycdrom
```

-v attiva la modalità verbose.
-o immagine.cd fa si che l'immagine prodotta sia contenuta nel file immagine.cd.
-R abilita le Rock Ridge.
-a include tutti i file
-m core esclude dall'immagine i file core (potete anche usare più di un parametro -m e usare i wildchar).
-V "My Volume" specifica l'etichetta di volume
-P "LFF spa" specifica la "casa" produttrice
-p "Pinco Pallino" l'autore.
/jobs/cdrom è la directory dalla quale parte la costruzione dell'immagine.

Quando dovete distribuire le vostre creazioni vi conviene abilitare le **Rock Ridge** con il parametro **-r** invece che con **-R**: in tal modo i permessi dei file vengono modificati ponendo sia gid che uid a 0, i file e le directory divengono leggibili da tutti, se uno dei bit execute è attivo vengono attivati tutti i corrispondenti bit ed infine tutti i permessi di scrittura sono azzerati. Se il vostro cd deve essere letto in ambiente **windows** è necessario attivare le **Joliet** con il parametro **-J** al posto di -r o -R.

Una volta ottenuta l'immagine ISO siamo pronti a trasferirla con il solito cdrecord:

```
cdrecord -v -dev=0,3,0 -speed=4
                 -data immagine.cd
```

Non abbiamo fatto altro che informare cdrecord di scrivere il contenuto del file immagine.cd sul masterizzatore come una traccia dati.

Se vogliamo copiare un cd la cosa **migliore** da fare è prelevare l'immagine iso dal cd stesso:

```
dd if=/dev/cdrom of=immagine.cd
```

a questo punto si può usare cdrecord come per masterizzare una qualsiasi immagine ISO9660.

Copia on-the-fly

Per copia **"al volo"** dei cd si intende la copia **diretta da cd a cd** su sistemi dotati di un masterizzatore e un cdrom. Il cd contenuto nel cd-rom viene copiato su quello nel masterizzatore **senza passare attraverso i file immagine** nell'hard disk. Cdrecord esegue la copia in questa modalità con il comando

```
cdrecord -v -dev=3,0
        -isosize /dev/cdrom
```

che funziona correttamente solo nel caso, molto frequente, che ci sia una sola traccia dati nel cd originе.

Non esiste **alcuna differenza** tra un cd creato mediante un'immagine e quello creato al volo tranne che il primo modo di procedere è più **affidabile** ma richiede una certa **quantità** di **spazio libero** sull' harddisk.

Cd misti...

Sempre più spesso vengono creati cd "misti",



Riquadro 3

X-CD-ROAST

L'URL dove è possibile prelevare l'ultima versione di x-cd-roast è:

http://www.fh-muenchen.de/rz/xcdroast

Il programma funziona con le librerie Tcl/Tk/Tix. Sebbene possiate avere le Tcl/Tk difficilmente possedete le Tix. Se avete coraggio, potete prelevare e compilare le Tix, ma scoprirete che è necessario avere i sorgenti anche delle Tcl/Tk. A questo punto puo' essere conveniente scaricare direttamente i binari all'url precedente.
Al momento x-cd-roast include la versione 1.6 di cdrecord ed usa cdparanoia come grabber per i file audio (mentre nell'articolo utilizziamo cdrecord 1.8 e cdda2wav); la cosa non crea problemi con cdrecord ma può dare qualche noia quando si preleva l'audio da un cd: vi consiglio di continuare a utilizzare la linea di comando quando è necessario copiare un cd audio oppure di rinunciare a cdda2wav e utilizzare l'altrettanto valido cdparanoia.

ovvero con **una traccia dati** e **una o più tracce audio** sullo stesso supporto. Questi casi possono essere gestiti **tranquillamente** con cdrecord. L'unica accortezza èquella di catturare le tracce audio con **cdda2wav** e quella dati con **dd**. Si può capire **subito** se un cd è "**misto**" dall'output del comando

```
cdda2wav -D0,1,0 -toc
```

che visualizza la tabella dei contenuti (toc) del cd inserito nel driver sul bus 0 con id 1 e lun 0. Supponiamo di aver catturato l'audio in audio.wav e i dati in dati.cd, il comando

```
cdrecord -v -dev=0,3,0
          -speed=4 -data dati.cd
                -audio audio.wav
```

esegue la scrittura sullo stesso supporto delle due tracce.

Test e finalizzazione

Può essere utile, soprattutto sui sistemi che non si conoscono o con I/O non molto stabile, effettuare un **test di scrittura** prima di passare alla masterizzazione vera e propria. Sebbene questo **non escluda errori nella fase di scrittura effettiva**, si può **ragionevolmente** escludere che si verifichi un **buffer underrun** nella fase di scrittura effettiva. Con cdrecord basta aggiungere il parametro **-dummy** per effettuare il test di scrittura; ad esempio

```
cdrecord -v -dev=0,3,0
  -speed=4 -dummy -data dati.cd
                -audio audio.wav
```

esegue il test sul cd misto visto prima.
Un'altra funzionalità interessante è quella di evitare che il cd venga **"finalizzato"** ovvero che vengano scritti la toc e altri dati necessari ai lettori per localizzare le varie tracce sul cd. Quando un cd non è finalizzato è **possibile aggiungere** altre tracce in coda a quelle già scritte (cosa che non è fattibile su un cd finalizzato); **per contro** un certo numero di lettori non riesce a leggere cd non finalizzati. Con cdrecord la finalizzazione viene disabilitata con il parametro **nofix**; ammettiamo di voler scrivere due tracce audio senza finalizzare il cd:

```
  cdrecord -v -nofix -dev=0,3,0
                          -speed=4
               -audio audio1.wav
               -audio audio2.wav
```

il disco così prodotto **normalmente non è leggibile** da un lettore di cd audio ma è possibile finalizzarlo con il comando

```
cdrecord -v -fix -dev=0,3,0
                          -speed=4
```

oppure è possibile aggiungere altre tracce:

```
  cdrecord -v -nofix -dev=0,3,0
    -speed=4   -audio audio3.wav
               -audio audio4.wav
```

X-CD-ROAST

Ricordare a memoria tutti i parametri necessari per una **corretta masterizzazione** può essere alquanto difficile. Se utilizzate l'ambiente X è interessante installare un **front-end grafico per** cdrecord. Tra quelli disponibili il migliore è senz'altro **X-CD-ROAST** (vedi riquadro 3). In figura 1 e 2 sono riportati gli screenshoot relativi: il suo utilizzo è **semplice se non banale** e non dovreste avere problemi se avete un po' di pratica con cdrecord.



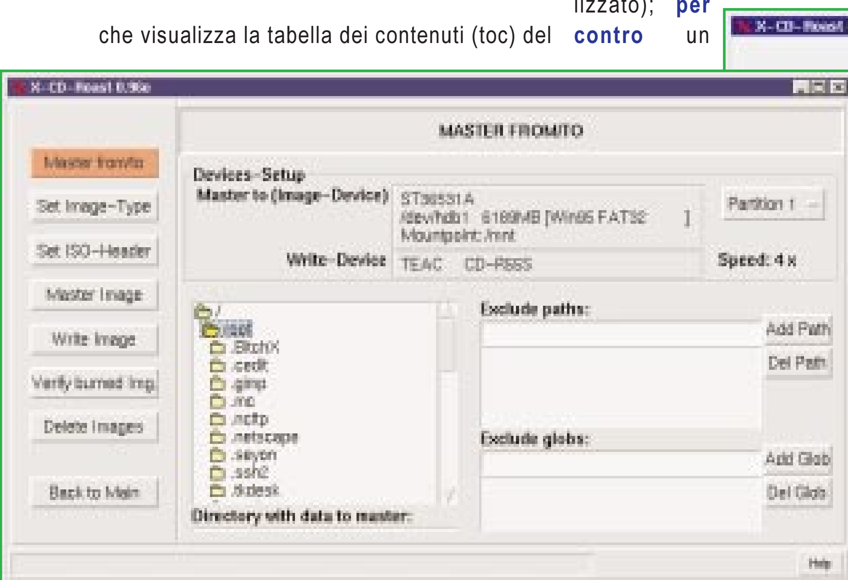


Figura 4: una schermata di impostazioni della masterizzazione. Più semplice? Questione di abitudine!



Figura 5: la schermata iniziale di X-CD-ROAST.

il nostro DATABASE sul World Wide Web

FILIPPO STEFANELLI
f.@oltrelinux.com

Webmaster di dislessici.org, si occupa di sicurezza di siti web e del loro interfacciamento con basi di dati.

Avete la necessità di portare il vostro database su internet oppure nella vostra intranet? Gli usi possono essere davvero tanti: un modulo d'ordine online, ad esempio, che controlli in tempo reale la giacenza, condividendo lo stesso database del magazzino. Analizzeremo una soluzione che utilizza Apache -server Web-, MySQL -server SQL- e PHP -creazione pagine dinamiche -.

Di pari passo allo sviluppo di internet vengono creati siti web sempre più **interattivi** e **accattivanti**. Per realizzare **pagine** stando al passo coi tempi ormai non basta più il solo **html** ma servono strumenti **nuovi** e molto **versatili** che, interagendo con il **server web**, offrono nuove opportunità di **sviluppo**. Nel nostro caso vedremo come utilizzare i dati contenuti in un **database** all' interno di un sito. Per far questo dobbiamo innanzitutto sapere cos'è un database e in che modo lo useremo.

SQL e MySQL

Il database che costruiremo e gestiremo è di tipo relazionale. Un database di questo genere contiene uno o più oggetti che prendono il nome di tabelle. Ogni tabella ha un nome **univoco** grazie al quale è distinta dalle altre tabelle dello stesso database.I dati che noi immettiamo sono contenuti in queste tabelle attraverso un reticolo di colonne e righe. Quest' ultime contengono il dato vero e proprio (**record**) mentre le colonne identificano la classe di dati che in essa è contenuto. Ecco un piccolo esempio:

NOME	COGNOME	ETA'	SESSO
mario	rossi	35	m
carla	bianchi	30	f
silvio	verdi	31	m

Com'e' facilmente intuibile le colonne sono NOME,COGNOME,... mentre i records sono le 3 righe sottostanti.
Come possiamo fare ad immettere e a selezionare tutti questi dati all' interno del nostro database? Attraverso il **linguaggio** dei **database relazionali** che si chiama **SQL** (*Structured Query Language - inizieremo a breve un corso di questo linguaggio in uno dei prossimi numeri*). Analizzeremo la sintassi e l' uso di questo potentissimo strumento più avanti. Resta da scegliere quale database utilizzare. Ve ne sono diversi di grande qualità che trattano i dati con SQL. Data la sua d**isponibilità per linux**, l' estrema **velocità** e **stabilità** ho scelto MySQL prodotto dalla T.c.X. Per far interagire il database con un sito in base alle richieste di un utente serve una "colla" **potente** ed **efficente** in grado di attaccare a **MySQL** e al **server WEB**.

> «Per realizzare pagine stando al passo con i tempi ormai non basta più il solo html, ma servono strumenti nuovi e molto versatili...»

PHP e APACHE

Per far lavorare insieme il server Web e il server SQL utilizzeremo **PHP**, un **versatilissimo** linguaggio **embedded** nell' html. L' utilizzo di PHP è aumentato **incredibilmente** nell' ultimo anno; un sondaggio non ufficiale condotto da Netcraft ha evidenziato che il numero di host che usano PHP è passato dai **7.500** del Giugno '98 ai **410.000** di Marzo '99. Il modello di funzionamento di PHP è molto semplice: quando al server web viene richiesta una pagina scritta in **PHP** (che si contraddistingue per l' estensione .PHP3) questo cerca nella pagina il codice in PHP e lo **processa eseguendo** ciò che vi è scritto, una volta finita questa procedura restituisce all' utente finale una pagina scritta interamente in **html**. Grazie a questa "**collaborazione**" si aprono le strade per nuove e interessanti soluzioni che analizzeremo più avanti. Anche il server web è ugualmente **importante**. Se non si ha già a disposizione un server con il **supporto** per PHP allora questo deve essere installato e configurato in modo che riconosca come tali le pagine scritte in PHP. Il server web che ho scelto di utilizzare è **Apache**, per noi utenti linux è praticamente una scelta obbligata data la sua stabilità e le sue ottime prestazioni.
In alcune **distribuzioni** Apache è già pronto all' uso e **configurato** per **riconoscere** PHP, in caso contrario dobbiamo **scaricarlo** e **installarlo** insieme agli altri due programmi. I link per scaricare le ultime versioni aggiornate di questi programmi si trovano nel secondo riquadro al termine di questo articolo.
A questo punto siamo pronti per iniziare.

Installazione e configurazione

Un piccolo consiglio prima di incominciare, è buona norma leggere sempre attentamente i file README e INSTALL che accompagnano questi programmi in quanto possono evitare di **perdere** molto tempo prezioso.
Una volta scaricati tutti e tre i programmi iniziamo a scompattare e a compilare (senza utilizzare alcun binario precompilato) per primo MySQL dato che non dipende strettamente da nes-



suno degli altri due.
Una volta spostato il file nella directory dove va scompattato eseguiamo questo comando

```
tar -xvzf mysql-x.xx.xx.tar.gz
```

(x.xx.xx è il numero della versione del programma)
A questo punto ci ritroveremo con una nuova **directory** chiamata **mysql-x.xx.xx**, entriamo al suo interno

```
cd mysql-x.xx.xx
```

e prima di lanciare lo **script di configurazione** (**configure**) controlliamo le opzioni che offre digitando

```
./configure --help
```

per vedere la lista di possibli parametri. Può essere molto utile installare tutto MySQL all' interno di una **stessa directory**, in questo modo si tengono sotto controllo tutti i file che compongono il programma. Questo può evitare molta **confusione** ad un utente inesperto. Il comando che va scritto per far questo è:

```
./configure --prefix /usr/local/mysql
```

che installa tutto MySQL in /usr/local/mysql. Finito lo script di configurazione lanciamo il comando

```
make
```

ed infine

```
make install
```

per **installare** tutti i files appena compilati nella locazione che abbiamo scelto prima.

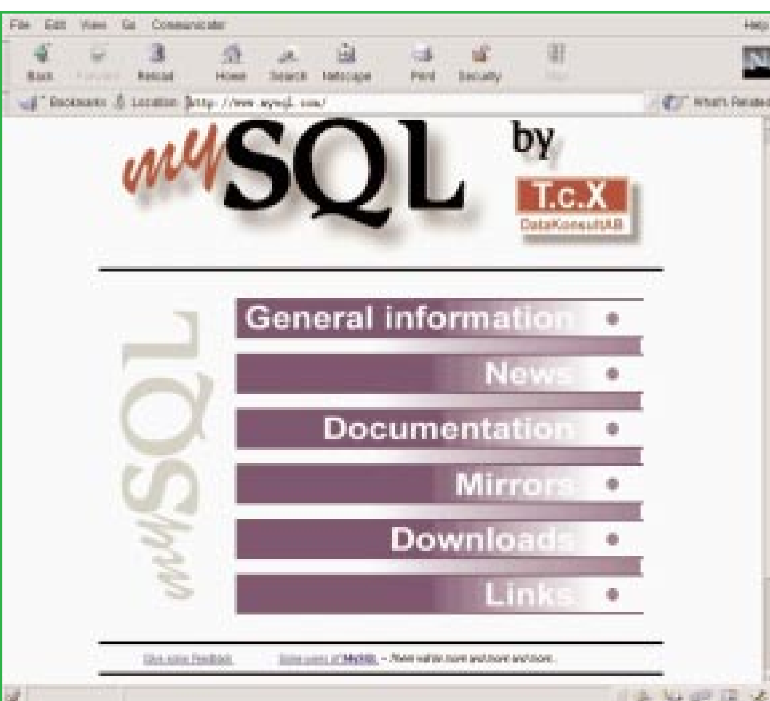


Figura 1: Sito di MySQL raggiungibile a www.mysql.com



Figura 2: Home page PHP a www.PHP.net

Si possono **evitare** tutte queste operazioni procurandosi un binario precompilato che va cercato all' interno del sito ftp.
Ora è il momento di installare i **permessi di default** (quelli dell' utente **root** di MySQL che **amministra** tutto il programma e assegna i **diritti** di **scrittura** e **lettura** sulle tabelle in un modo molto simile al root di un sistema linux con gruppi e utenti ai quali dare i permessi sui file) per far questo lanciamo il seguente comando (dalla directory dove abbiamo installato MySQL)

```
bin/mysql_install_db
```

fatto questo se vogliamo lanciare il server MySQL basta scrivere

```
bin/safe_mysqld &
```

attraverso il programma **mysqladmin** (bin/mysqladmin) si può, fra le altre funzioni di **amministrazione**, **fermare** o **controllare** lo stato del server. Inoltre può essere comodo impostare MySQL in modo che **venga lanciato automaticamente** al **boot** di Linux, per far questo copiate il file **mysql.server**, che si trova nella directory **support-files** dell' installazione, nella directory di **init** del vostro sistema.

A questo punto è il momento di configurare **correttamente** PHP e Apache. Una volta scaricati dobbiamo per prima cosa scompattarli

```
tar -xvzf php-3.x.xx.tar.gz
tar -xvzf apache_1.x.x.tar.gz
```

ora entriamo nella directory di Apache

```
cd apache_1.x.x
```

e iniziamo con le procedure dell' installazione che sono descritte molto bene nel

Figura 3: Il file di configurazione di PHP: PHP3.ini

file INSTALL di PHP.
Come abbiamo già visto in precedenza ci sono alcune opzioni per il **configure** che vanno specificate, in questo caso

```
./configure --prefix=/usr/local/apache

cd ../php-3.x.xx

./configure --with-mysql=/usr/local/mysql
--with-apache=../apache_1.x.x

make

make install

cd ../apache_1.x.x

./configure --prefix=/usr/local/apache
--activate-module=src/modules/php3/libphp3.a

make

make install
```

adesso copiamo il file di configurazione di php (php3.ini) nella sua locazione di default

```
cd ../php-3.x.xx

cp php3.ini-dist /usr/local/lib/php3.ini
```

e per finire dobbiamo scommentare (togliere il "#") le seguenti righe dal file httpd.conf che si trova in /usr/local/apache/conf

```
AddType application/x-httpd-PHP3 .PHP3

AddType application/x-httpd-PHP3-source .PHPs
```

in questo modo Apache riconosce le estensioni delle pagine scritte in PHP.
Prima di far partire il nostro server nuovo fiammante un paio di **considerazioni** sulla **sicurezza** in PHP. Come si può vedere dalla Figura 3 le opzioni che permettono di configurare PHP sono **molte** e proprio per questo consentono di avere un ottimo controllo su quello che il **programma dovrà o non dovrà fare**. **In un programma di questo genere, che può leggere, modificare ed eseguire i file del computer, l' aspetto della sicurezza non può essere tralasciato, come ci dimostra anche l' importanza che il manuale attribuisce all' argomento** (vedere Figura 4). Vi sono numerose opzioni che riguardano la sicurezza; grazie a questi settaggi si riesce ad ottenere un **buon compromesso libertà/sicurezza** che va adattato in base alle esigenze di ogni utente. Un opzione che consiglio di abilitare, attraverso il php.ini, e' il "**safe_mode**". Quando questa modalità è attiva PHP lavora solo all' interno di una determinata zona che noi specifichiamo attraverso l' opzione "**doc_root**"; in questo modo la zona di lavoro è limitata ed **eventuali danni** sono più **circoscritti**. Per qualsiasi altra informazione si può consultare il capitolo 3 del manuale.

Prima pagina: Hello, World!

Per iniziare a scrivere la nostra prima pagina con PHP dobbiamo innanzitutto lanciare **Apache**. Quindi eseguiamo il comando

```
bin/apachectl start
```

dalla directory dove e' stato installato.
Di default Apache cerca i file che verranno visualizzati nel web nella directory **htdocs**, quindi è lì che andranno tutte le pagine che editeremo.
Creiamo un file **primapagina.php3** che contiene queste righe:

```
<html>
<header>
<topic> La nostra prima pagina con PHP </topic>
</header>
<body>
<?PHP echo("Hello, world!"); ?>
```

Riquadro 1

```
<?PHP ... ?>
<script language="PHP"> ... </script>
<? ... ?>
<% ... %>
```

Le ultime due modalità devono essere abilitate tramite i parametri ***short_open_tag*** e ***asp_tags*** che devono essere settati "On" nel PHP3.ini.

Come racchiudere delle istruzioni PHP in una pagina html.



```
</body>
</html>
```

"<?PHP" e "?>" sono i **tags** di **apertura** e **chiusura** di una qualsiasi istruzione PHP, vi sono anche altri modi di racchiudere del codice PHP che sono elencati nel primo riquadro.
Ora andando semplicemente con un qualsiasi browser all' indirizzo http://loalhost/primapagina.php3 vedremo la nostra prima piccola magia. Infatti la pagina che si vede contiene solo la scritta "Hello world!" e guardando il sorgente di questa pagina si vedrà che le istruzioni scritte in PHP sono completamente scomparse. Ora proviamo un altra "feature" utilissima e cioè la possibilità di vedere il "sorgente" vero e proprio di una pagina scritta con PHP. Prima abbiamo specificato all' Apache di riconoscere anche l' estensione .phps, quindi ora basta semplicemente modificare l' estensione di primpagina.php3 in primapagina.phps e in questo modo, dopo aver caricato quest' ultima con un browser, tutto il codice scritto in PHP risulterà colorato in modo diverso a seconda che si tratti di una funzione, tag, testo o un qualsiasi altro parametro. Per la nostra semplice pagina non ha molto senso ma quando si devono analizzare numerose righe scritte in PHP diventa veramente molto utile, come si può intuire guardando la Fig. 5.

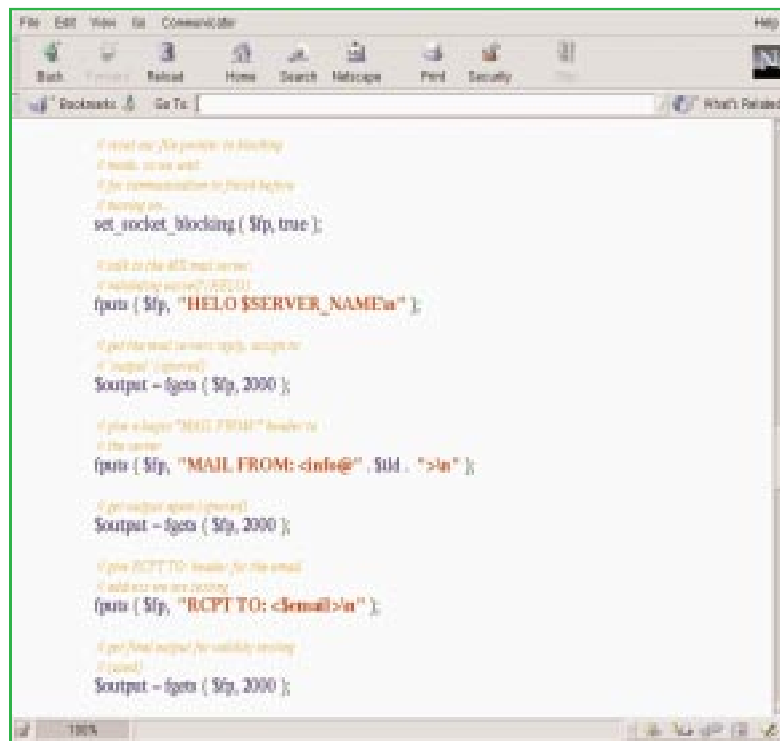
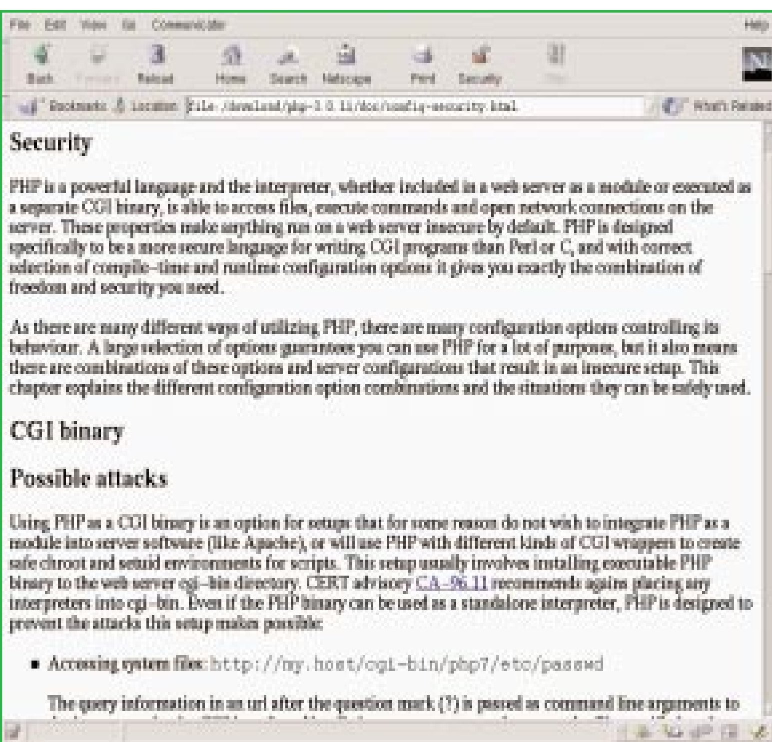


Figura 5: Un esempio di come il codice colorato può facilitare la lettura del PHP.

Conclusioni

In questa prima puntata abbiamo evidenziato i **principi generali** di funzionamento dei database di tipo relazionale. Abbiamo visto cos'è e in che modo **funziona PHP** e come installare e far funzionare sul nostro computer: **Apache, PHP e MySQL**. Inoltre abbiamo affrontato gli aspetti basilari di quella che può essere la **sicurezza** in un ambiente dove viene usato PHP. Infine abbiamo scritto la nostra prima pagina in PHP.
Sul **prossimo numero** vedremo molti altri **comandi PHP** e inizieremo a immettere e a trattare dati con **MySQL** attraverso il linguaggio **SQL**, in questo modo partiremo con la realizzazione del nostro sito **interattivo!!!**. Dunque non mancate!



File Edit View Go Communicator Help

Location: file:/download/php-3.0.11/doc/config-security.html

Security

PHP is a powerful language and the interpreter, whether included in a web server as a module or executed as a separate CGI binary, is able to access files, execute commands and open network connections on the server. These properties make anything run on a web server insecure by default. PHP is designed specifically to be a more secure language for writing CGI programs than Perl or C, and with correct selection of compile-time and runtime configuration options it gives you exactly the combination of freedom and security you need.

As there are many different ways of utilizing PHP, there are many configuration options controlling its behaviour. A large selection of options guarantees you can use PHP for a lot of purposes, but it also means there are combinations of these options and server configurations that result in an insecure setup. This chapter explains the different configuration option combinations and the situations they can be safely used.

CGI binary

Possible attacks

Using PHP as a CGI binary is an option for setups that for some reason do not wish to integrate PHP as a module into server software (like Apache), or will use PHP with different kinds of CGI wrappers to create safe chroot and setuid environments for scripts. This setup usually involves installing executable PHP binary to the web server cgi-bin directory. CERT advisory CA-96.11 recommends agains placing any interpreters into cgi-bin. Even if the PHP binary can be used as a standalone interpreter, PHP is designed to prevent the attacks this setup makes possible:

- Accessing system files: http://my.host/cgi-bin/php?/etc/passwd

The query information in an url after the question mark (?) is passed as command line arguments to

Figura 4: Il capitolo del manuale di PHP dedicato alla sicurezza

Bibliografia

[1] "PHP3 Manual", PHP Documentation Group.
[2] "MySQL Reference Manual for version 3.22.25", www.mysql.com



Riquadro 2

Elenco dei mirror italiani.

Per MySQL:
http://www.teta.it/mysql/Downloads/MySQL-3.22/

Per PHP:
http://PHP.fmal.com/distributions/
http://it2.PHP.net/distributions/

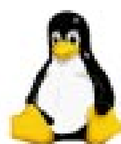
Per Apache:
http://Apache.phenixnet.it/dist/
http://Apache.sindata.it/
ftp://Apache.sindata.it/pub/Apache/dist/

Dove trovare le ultime versioni e i binari precompilati dei programmi.

Visitateci alla smau 99
Pad. 25 Sal.I Stand B06B

▸ Internet ▸ Rete
▸ Office ▸ Giochi ▸ Sound ▸ Video
e tanto, tanto di più ...





In tutto più di 1.300 applicazioni!

Linux facile con SuSE Linux 6.2

Con YaST, il tool di SuSE, i novelli di Linux riusciranno ad installare e configurare facilmente il loro sistema. Con più di 1300 applicazioni – dal pacchetto Office, alle banche dati fino all'MP3-Player – riceverete una raccolta attuale di tutti i migliori programmi liberi di Linux. Scegliete fra i diversi ambienti desktop e create il Vostro desktop personale. Nell'ampio manuale troverte utili e provati consigli e suggerimenti. Da non dimenticare i 60 giorni di supporto per l'installazione!

€ 46,-

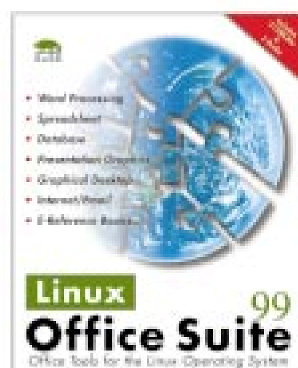
Ora con 6 CD-ROMs & 420 pagine di manuale

Linux Office Suite 99

Sperimentatene la produttività sul Vostro desktop! SuSE Vi presenta una raccolta di Office-Tools che appagherà completamente i Vostri desideri: **Applixware 4.4.2** offre l'elaborazione di testo, il foglio elettronico e la presentazione, come pure un facile collegamento alle banche dati SQL e ambiente di sviluppo oggetto-orientato: il tutto con superfici grafiche! Idoneo per tutte le distribuzioni Linux conosciute!

€ 75,-

2 CD-ROMs e manuale inglese

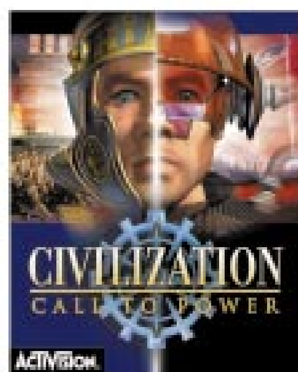


CIVILIZATION: Call To Power

CIVILIZATION: con Call To Power proverete l'assoluto piacere dei giochi Linux. La terza versione di questo classico fra i giochi strategici Vi entusiasmerà grazie alle sue nuove unità, edifici e meraviglie. Per la prima volta, sarete in grado di sconfiggere i Vostri nemici senza usare armi, ma strategie ecologiche o religiose. Affascinanti sono la grafica realistica e la possibilità di gioco per più persone. 7000 anni sono nelle Vostre mani! Realizzate il Vostro mondo a seconda dei Vostri desideri: partendo da un'epoca primitiva fino ad un mondo futuristico e guidate il Vostro popolo al benessere a alla contentezza.

€ 49,-

1 CD-ROM e manuale inglese



Fate ancora oggi le Vostre ordinazioni presso

Italsel srl
Via Lugo 1
40128 Bologna
Tel. 051 32 04 09
Email: prossi@italsel.com

ATHENA intern. Bookshops
Via G. Campi 284/a
41100 Modena
Tel. 059 37 08 42
Email: athena@mondolinux.com



SuSE GmbH
Schanzäckerstraße 10
D-90443 Nürnberg
Tel. +49 911 740 53 81
Fax +49 911 741 77 55
Email: suse@suse.de

http://www.suse.it/i



NetQuiz

soluzione client/server Windows/Linux Delphi/Interbase

Jilani Khaldi - j.khaldi@oltrelinux.com

La grande trovata di questo fine secolo è senza dubbio Internet. Da uno strumento per élite si è trasformato in un fenomeno di massa, estendendosi su tutto il pianeta con una rapidità di diffusione mai raggiunta da nessuno degli altri fenomeni di massa che l'hanno preceduto: la radio, il telefono, la televisione…
Questa impetuosa avanzata di Internet sta influenzando molte delle nostre abitudini di vita e di lavoro. Tale influenza è ancora più sentita nella comunità degli sviluppatori di software, quando si ipotizza che in un futuro non molto lontano, Internet sarà la piattaforma d'eccellenza per lo sviluppo del nuovo software; o almeno così si crede. In questa ottica, molti giganti del software stanno incrementando notevolmente la propria fortuna, grazie ad una semplice previsione. Le loro proposte oggi, variano dalle semplici estensioni ai "vecchi" linguaggi di programmazione, ai sofisticati strumenti nuovi di zecca, appositamente disegnati per lo sviluppo Web based.

Web based vs. Client/Server

Anche se le nuove applicazioni che usano le moderne tecnologie Internet, sono ancora di tipo C/S (*un client che si connette ad un server e riceve delle informazioni*), esse hanno una caratteristica del tutto particolare. Il browser Web ha sostituito la classica applicazione client; e quella server è diventata un insieme di applicazioni. Almeno due. Una è il Web server, come **IIS** della Microsoft, **JWS** della Sun, **Apache** di Apache.org, **Jigsaw** di W3C.org…, l'altra invece, è un'applicazione sviluppata dal programmatore che fa da tramite tra il Web server e il database.
Se le applicazioni Web based sono delle ottime soluzioni per il commercio elettronico, la pubblicazione e l'accesso ai dati tramite Internet/Intranet, quelle classiche C/S sono le più adatte per le applicazioni gestionali, come quelle bancarie ed assicurative.
Con tutte le qualità e le novità che le applicazioni Web based hanno, esse presentano ancora delle lacune difficili da colmare a breve termine. Una di queste è la non persistenza delle connessioni. Quando navighiamo sul sito A e passiamo in seguito sul sito B, perdiamo la connessione col sito A. Questo limite è dovuto al protocollo HTTP, sul quale si basa questo genere di applicazioni, e che non è stato del tutto risolto neanche con l'ultima versione (1.1). Un altro problema, non meno serio, è il ricorrere ad un applet Java oppure ad oggetti ActiveX per ambienti Windows, per avere una ristretta elaborazione locale. Questi problemi sono praticamente inesistenti nelle classiche applicazioni C/S, dove tutta la potenza del client viene sfruttata ed esaltata.

> «La nostra combinazione è Delphi/Interbase. Una sinergia difficile da battere. Semplicità, potenza, eleganza.»

Sviluppare un'applicazione C/S

Lo scopo di questo articolo e dei prossimi, è proporre al lettore delle soluzioni pratiche, per permettergli in seguito di provarle e adattarle alle proprie esigenze. Quindi, l'obiettivo da raggiungere è lo sviluppo di un'applicazione C/S 2-Tier completa. La scelta di questa tipologia di software riflette pienamente tale scopo, visto il significativo ruolo che hanno tali applicazioni nella nostra quotidianità. Esse attualmente dominano il mondo. Basta pensare ai nostri dati anagrafici, conti bancari, targhe delle nostre automobili… Sono quasi tutte applicazioni C/S 2-Tier.
Le ragioni della loro grande diffusione sono veramente tante. Le applicazioni C/S sono molto facili da realizzare e richiedono poche risorse per essere sviluppate, grazie agli ottimi strumenti di sviluppo disponibili sul mercato. Sono adatte per la gestione delle piccole e medie imprese, diffusissime in Italia e per conseguenza, sono le più richieste.

NetQuiz

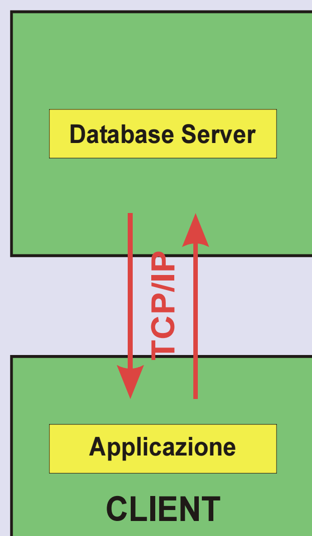
E' il nome della nostra applicazione. Come suggerisce il nome, è un'applicazione C/S per creare, e svolgere quiz in rete. Sarà di grande utilità. Basta pensare all'introduzione della prova a quiz all'esame di maturità e come si diffonderà d'ora in poi questa modalità di valutazione, ai sondaggi, ai test di selezione, all'apprendimento passivo.

Ingredienti

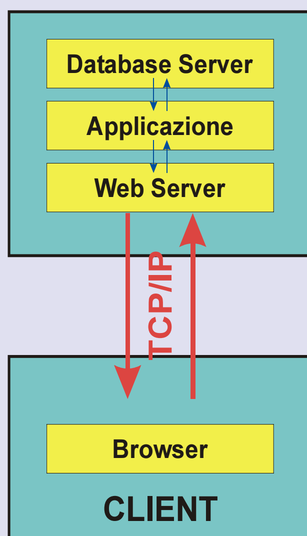
Per lo sviluppo di un'applicazione C/S occorre una rete di computer, composta almeno da due PC (un client e un server) collegati fisicamente tra loro tra-



Figura 1: La fase di collegamento di un utente con il server.



Schema di una classica applicazione client/server 2-Tier



Schema di una applicazione WEB

mite **cavo** e **schede** di **rete**, un sistema operativo adatto per il **networking** che sarà installato sul server, un sistema operativo sul client, un **database server** e un **linguaggio** di **programmazione** per scrivere l'applicazione.

Per quanto riguarda i sistemi operativi è ben presto fatto. **Linux** lato **server**, **Windows** (da 95 in su) lato **client** e il **TCP/IP** come **protocollo di comunicazione.**

Per il database server, abbiamo un'ampia scelta. Dai nomi altosonanti, più o meno costosi: Oracle, DB2, Sybase, Informix, SQL Server, Interbase, ..., fino ad arrivare a quelli free, più o meno noti e usati: mSQL, MySql, Postgres, ... Gli strumenti di sviluppo sono altrettanto validi e numerosi. Visual Basic, Delphi, Power Builder, Visual C++, C++ Builder, ambienti Java, ...

Quale scegliere? Come vedete, il calcolo combinatorio ci dice che le soluzioni sono **tantissime**, almeno matematicamente parlando. In realtà, le cose stanno diversamente. Per ogni **sviluppatore**, le scelte sono veramente ridotte. Anzi, sono addirittura solo una o due.

Chiediamolo a Carlo V

Quando chiesero a Carlo V come adoperava tutte le lingue che conosceva, egli rispose: "parlo spagnolo a Dio, italiano alle donne, francese agli uomini e tedesco al mio cavallo".

Battuta a parte, ogni sviluppatore che ha familiarità con vari linguaggi di programmazione, ambienti di sviluppo e database, sa che la scelta dello strumento di sviluppo è strettamente legata al **tipo di applicazioni** da realizzare. Quindi, sceglie sempre lo strumento di sua conoscenza più adatto per tale scopo. A meno che non gliene venga imposto uno diverso.

Allora diciamolo subito. La nostra combinazione è: **Delphi/Interbase**. Una sinergia difficile da battere. E' una combinazione di **semplicità**, **potenza** ed **eleganza**. D'ora in poi, **tutti** gli sviluppatori Delphi sono chiamati in causa. Ma anche tutti gli altri colleghi che usano strumenti di sviluppo diversi sono chiamati a partecipare. La codifica avverrà in due parti distinte: quella client e quella server. Quest'ultimo, essendo basata esclusivamente su procedure SQL standard, la scelta del database **non è affatto vincolante** e rimane una scelta del tutto **personale**. Per quanto riguarda la parte client, prima di ogni codifica in Delphi, la porzione di codice da sviluppare sarà presentata in **pseudo linguaggio**, per permettere ad ogni programmatore di scriverla in seguito nel suo **idioma preferito**.

> «Finito NetQuiz, si procederà con WeQuiz. Tutto il lavoro sarà integralmente riutilizzato!»

I simpatici esclusi: MySQL e Postgres

MySQL sarebbe veramente un'ottima soluzione per tale tipo di applicazioni. E' **velocissimo**, **free** e **adatto a immagazzinare grandi quantità di dati**. Purtroppo **non** gestisce le **transazioni**. Ogni applicazione C/S che si rispetta deve, e sottolineo deve, gestire le transazioni.

E Postgres? Che io sappia, non ci sono ancora i driver per accedere ai dati da Delphi, altrimenti sarebbe **una valida alternativa**.

Comunque, tengo a precisare che Interbase è un software di qualità industriale e la sua affidabilità, potenza e sicurezza sono davvero notevoli.

NetQuiz in breve

Nome dell'applicazione: NetQuiz
Descrizione: Applicazione C/S per creare e fare i quiz rete
Piattaforma server: Linux (tutte le distribuzioni)
Piattaforma client: Windows 95/98/NT
Database SQL: Interbase 4.2 (free per Linux: http://www.interbase.com)
Ambiente di sviluppo: Delphi 2, 3, 4

Anatomia di NetQuiz

Come tutte le applicazioni C/S (2-Tier), NetQuiz sarà composta da un'**applicazione client**, sviluppata in Delphi su Windows (vedi prototipo in figura) e un'**applicazione server**, consistente nel database con tutte le procedure SQL su Linux. L'utente, dopo essere stato **autenticato**, si **connette** al server, **sceglie** il quiz che vuole svolgere (ci sarà una grande varietà) con le opzioni che desidera attivare. Dopo aver risposto alle domande, premendo un bottone, avrà il risultato e altre informazioni utili (tempo impiegato, numero degli errori, le risposte corrette…). Tutto dipenderà ovviamente dalle opzioni messe a sua disposizione e dai privilegi concessi.

Non finira' li'

Finito **NetQuiz**, si procederà con **WebQuiz**, che sarà la versione Web based di NetQuiz.
Ovviamente **tutto il lavoro** svolto lato server verrà **integralmente utilizzato**. Anzi, in futuro sia NetQuiz che WebQuiz continueranno ad avere la **stessa base di dati**, con tutti i vantaggi che ne derivano.

La prossima puntata inizieremo sul serio. Non mancate.



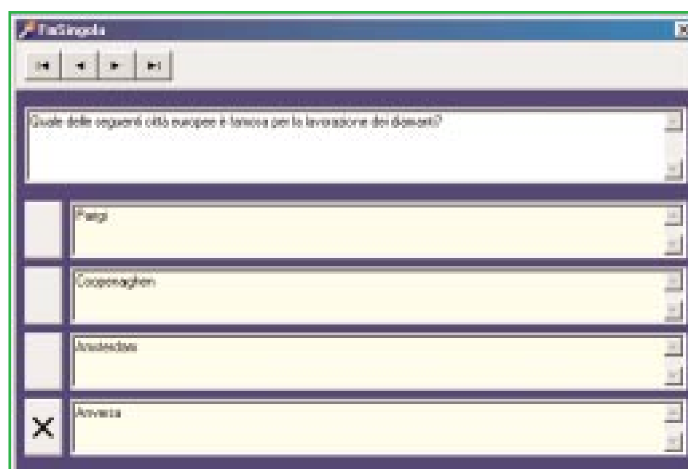


Figura 2: una maschera-esempio (beta) di un quiz.

LIBERI DI SCRIVERE
FINALMENTE TeX

Riccardo Murri
r.murri@oltrelinux.com

Laureando in Matematica, si interessa al mondo Linux da diversi anni. Oltre al Tex, è appassionato di HTML.

Quante volte per scrivere la nostra relazione contenente poche formule ci siamo scontrati con strumenti poco adatti e difficili da gestire. Il TeX è nato proprio per la pubblicazione di testi scientifici, ed è uno standard di fatto se si considera che quasi tutti i programmi di calcolo permettono di esportare i risultati in questo formato.

Nel lontano **1977**, Donald E. Knuth decise che non ne poteva più di vedere i suoi articoli martoriati dai tipografi delle riviste scientifiche: venivano introdotti più **errori** in fase di **composizione** di quanti non ne venissero corretti durante la **revisione** delle bozze; era giunto il momento che l'informatica venisse a riconciliare autori, editori e tipografi... Cinque anni dopo, con la pubblicazione di ``**Computers and Typesetting**", nasceva il sistema **TEX**, oggi il più usato programma di scrittura e composizione tipografica di **testi scientifici**.
TEX converte un file di testo in un file **.dvi**, pronto per la stampa; si possono inserire speciali comandi per istruire TEX sulla resa tipografica di alcune porzioni di testo, di più, TEX è un vero e proprio linguaggio di programmazione con cui si possono preparare complesse **macro** per la manipolazione e la **composizione** del testo. Molti pacchetti di macro sono stati sviluppati per facilitare la scrittura di documenti col TEX; il primo è certamente **Plain-TEX**, dovuto allo stesso Knuth, ma il più famoso è **LATEX**, scritto da Leslie Lamport alla fine degli anni '80, ed oggi arrivato alla versione LATEX2€

Cos'e' TEX?

TEX non è un programma per scrivere testi, ma per comporli **tipograficamente**. Mentre un comune word-processor vuole offrire una soluzione di scrittura per tutte le **esigenze**, dal creare un bigliettino di auguri al redarre una raccolta di poesie, TEX si **concentra** (e raggiunge il suo meglio) nella composizione di **testi scientifici** o comunque **fortemente strutturati**: articoli, relazioni o interi libri.
Per scrivere un file .tex si usa Emacs, vi o il vostro **editor** preferito; poi, TEX sostituisce (per quanto può una macchina) il lavoro del tipografo: arrangia i caratteri in righe, e queste in pagine, inserisce le figure nel testo e compone le note a pie' di pagina; TEX permette di variare un gran numero di **parametri** che controllano questa composizione: le **dimensioni** della pagina, la **grandezza** del carattere, la **spaziatura**, ecc. Tuttavia, TEX assumerà sempre di star componendo un libro, o un articolo di rivista, o comunque qualcosa che deve seguire delle rigide regole di impaginazione per assicurarne la migliore leggibilità: è difficile convincere TEX a impaginare poesia futurista, mentre mi sono sufficienti solo 4 comandi per comporre questo articolo su 2 colonne. TEX non è **WYSIWYG** Il lavoro di TEX si articola in più fasi distinte e separate:

1. si scrive un file di testo con estensione .tex con Emacs (vedi figura 1) o vi, o il vostro editor preferito
2. TEX trasforma il file .tex in un file binario .dvi (**device independent**);
3. si fa un'anteprima sul video del file .dvi con il programma xdvi (vedi figura 3);
4. il file .dvi viene convertito da **dvips** in un file POSTSCRIPT per stamparlo. (Altri convertitori sono disponibili per chi non ha stampanti POSTSCRIPT.)

Non è così complicato come può apparire a prima vista: per esempio, **Emacs** ha un modo (chiamato AUC-TEX) che permette di **controllare** da un menu **tutte le fasi della produzione**, anteprima e stampa di un documento .tex; altri editor hanno funzionalità **simili**, ed il programma **KLyX** permette di modifi-



Figura 1: Una schermata di XEmacs, durante la scrittura di questo articolo. XEmacs e GNU Emacs hanno un modo specifico per i file TEX, che permette di controllare la produzione, l'anteprima e la stampa del file senza uscire dall'editor, oltre ad una serie di macro per accelerare la scrittura.



Figura 2: Anteprima di questo articolo col programma kdvi, il programma di visione .dvi del KDE. Il programma xdvi ha le stesse funzionalità e gira su qualunque sistema X/Windows.

care file .tex in un ambiente **WYSIWYG**, pur conservando TEX come strumento di produzione del file da mandare in stampa.

TEX è un linguaggio di programmazione. TEX consente l'espansione di macro, costrutti condizionali, lettura e scrittura di file, operazioni aritmetiche, oltre ad una serie di routine specifiche per il controllo della resa tipografica. Questo permette di creare automaticamente sommari, indici analitici e delle figure, bibliografie; di includere nel testo tabelle di dati ed immagini create con programmi esterni...

TEX compone formule matematiche. Creato principalmente per scrivere articoli scientifici, TEX permette di mischiare al testo formule matematiche; di più, per scrivere le formule bastano solo i normali caratteri **ASCII**, e la sintassi è in effetti simile alla pronuncia (inglese) della formula (vedi figura 3). Questa è probabilmente la caratteristica principale di TEX, che lo distingue da programmi analoghi come groff o da sistemi di word-processing: non occorrono programmi esterni per comporre una formula, che così fa parte **integrante** del **testo** e non viene trattata come **un'immagine esterna**, e la **qualità tipografica** è decisamente professionale.

Inoltre, programmi come **MATHEMATICA** o **MAPLE V** permettono di salvare i risultati dei loro calcoli come formule nel formato TEX, pronte per essere incollate dentro una relazione.

TEX è portabile. Grazie al suo approccio a fasi separate alle preparazione di un documento, i singoli componenti di un sistema TEX sono più semplici e portabili; esistono eseguibili di TEX per la maggior parte dei sistemi operativi moderni, e TEX gira sui più vecchi **80286** come sulle più ambiziose **workstation**. Inoltre, un file .tex si scrive usando solo i caratteri **ASCII** (anche se dovete scrivere lettere accentate o caratteri indiani...) ed è quindi leggibile su ogni computer...

TEX è software libero. Se volete, tutte le sorgenti di TEX e LATEX sono disponibili per essere modificate alle vostre esigenze...

Cos'e' LATEX?

TEX è un linguaggio di programmazione, e LATEX è un programma che permette di utilizzare questo linguaggio. Lo scopo dichiarato di LATEX è di fornire un sistema di **preparazione** dei documenti per la stampa, che permetta di **separare** quanto più possibile la preoccupazione per la resa tipografica dall'organizzazione della scrittura. TEX ambisce a mettere a disposizione degli autori gli strumenti dei tipografi, e LATEX cerca di sostituire questi ultimi per quanto possibile; però, laddove un tipografo umano saprebbe distinguere il risalto da dare alle varie parti di un testo, LATEX ha bisogno di essere **guidato** da particolari **comandi** che indicano come il testo è suddiviso in capitoli, sezioni e sottosezioni, oppure se un certo gruppo di righe è un listato di un programma (che deve perciò essere reso in un carattere diverso da quello del testo corrente), ecc.

La caratteristica principale di LATEX è dunque di accettare in **ingresso** un file di testo segnato con pochi comandi che ne indicano la **struttura**, e produrre un documento pronto per la stampa, seguendo una composizione predefinita. L'idea alla base di tutto questo è che un autore deve essere libero di concentrarsi sul contenuto del testo, più che sul suo aspetto: LATEX farà pertanto ogni sforzo per adattare il vostro testo agli schemi di composizione tipografica professionale che ha in memoria. Ad esempio, con una **decina** di comandi LATEX potete scrivere un **documento**, **impaginarlo** su due colonne e **generare** automaticamente il **sommario**; se poi decidete che preferite l'impaginazione su una colonna sola, o la stampa fronte/retro, basta cambiare una sola riga ed il gioco è fatto.

LATEXadotta un sistema **modulare**, per cui si possono facilmente aggiungere estensioni che permettono di modificare il comportamento standard in varie maniere: per esempio esistono moduli per utilizzare i caratteri **POSTSCRIPT** oltre a quelli TEX, moduli per generare **bibliografie** in diversi formati, moduli per **disegnare** diagrammi, per includere listati di programmi, ecc.

$$W(\xi,h) = \log \int \exp \frac{1}{h}\Bigl(x\xi - \frac{x^2}{2} + \sum_{2(g-1)+n>0} \frac{a_{g,n}h^g x^n}{n!}\Bigr) \frac{dx}{\sqrt{2\pi h}}$$

```
W(\xi,h) = \log \int \exp \frac{1}{h} \Bigl( x\xi - \frac{x^2}{2} +
          \sum_{2(g-1)+n>0} \frac{a_{g,n}h^gx^n}{n!}\Bigr)
          \frac{dx}{\sqrt{2\pi h}}
```

Figura 3: Una formula matematica, ed il codice TEX necessario per produrla.

Un esempio di LATEX

Vediamo un po' un esempio di come può essere fatto un file LATEX; con il vostro editor preferito aprite un file esempio.tex e scrivete:

```
% esempio.tex
%
% Un esempio per dimostrare com'è fatto un file LaTeX
% (tutto ciò che segue '%' viene ignorato da TeX)
%
\documentclass{article}
\usepackage[latin1]{inputenc} % lettere accentate
\begin{document} % qui comincia il testo
\section{Una storia Zen}
% due paragrafi di testo ordinario...
Un uomo di AI attraversò le montagne per SAIL per vedere il
Maestro, Knuth. Quando fu arrivato, il Maestro non era da
nessuna parte. ``Dov'è il saggio chiamato Knuth?'', egli
chiese ad uno studente di passaggio.
``Ah,'' disse lo studente, ``non hai saputo. Egli è partito
```

```
in pellegrinaggio attraverso le montagne al tempio di AI per
cercare nuovi discepoli.''

Udendo ciò, l'uomo fu Illuminato.

\section{Una poesia... matematica.}

\begin{verse} % seguono versi, separati da '\\'
  Aleph-null bottles of beer on the wall, \\
  Aleph-null bottles of beer, \\
  You take one down, and pass it around, \\
  Aleph-null bottles of beer on the wall.
\end{verse}

\begin{verse}
  % $\aleph_0$ indica il simbolo matematico di Aleph-con-0
    $\aleph_0$ bottiglie di birra al muro, \\
    $\aleph_0$ bottiglie di birra, \\
    Ne prendi una, e la passi in giro, \\
    $\aleph_0$ bottiglie di birra al muro.
  \end{verse}
\end{document}
```

A questo punto, create il file **.dvi** col comando

```
latex esempio
```

e guardate il risultato con

```
xdvi esempio
```

Se tutto è andato bene, dovreste ottenere lo stesso **risultato** di figura 4.

Commentiamo un po' questo listato. Le **prime righe** (quelle che iniziano con '%') sono **commenti** e vengono ignorate da LATEX nel produrre il testo finale. Il file si apre con la dichiarazione

```
\documentclass{article}
```

che predispone LATEX a comporre un **articolo** (alternativamente, avremmo potuto scegliere un libro, una lettera, una relazione o dei lucidi). Segue

```
\usepackage[latin1]{inputenc}
```

per poter usare le lettere **accentate**, e non i soli caratteri **ASCII**.
Il testo vero e proprio comincia con

```
\begin{document}
```

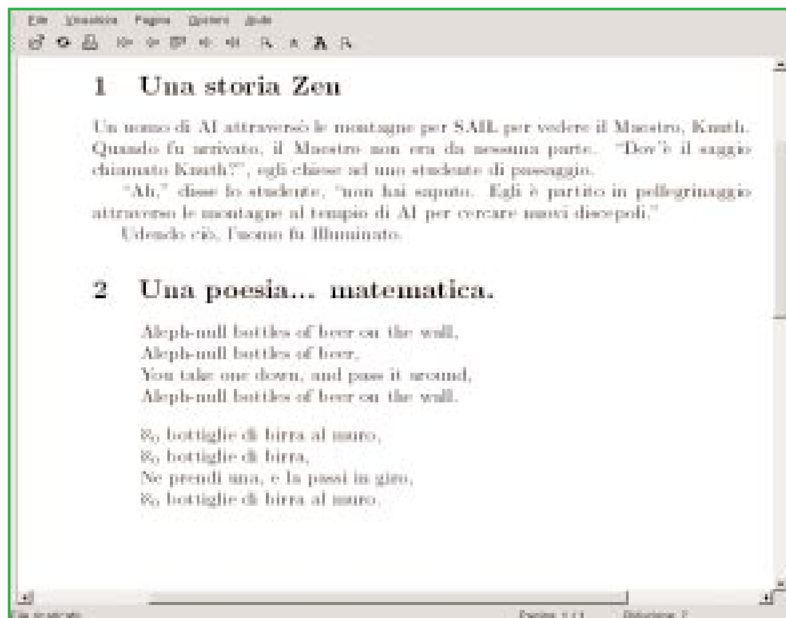
La riga

```
\section{Una storia Zen}
```

indica il **titolo** di una **sezione**, e seguono tre paragrafi di testo ordinario: notate come LATEX ignori gli **spazi** e la **suddivisione** in righe, e provveda comunque a fare una stampa **ordinata**. (cfr. Figura 2.1) . Infine,

```
\begin{verse}
....
\end{verse}
```

racchiudono del testo **poetico**, cioè diviso in versi, separati da '\\'.


1 Una storia Zen

Un uomo di AI attraversò le montagne per SAIL per vedere il Maestro, Knuth. Quando fu arrivato, il Maestro non era da nessuna parte. "Dov'è il saggio chiamato Knuth?", egli chiese ad uno studente di passaggio.
"Ah," disse lo studente, "non hai saputo. Egli è partito in pellegrinaggio attraverso le montagne al tempio di AI per cercare nuovi discepoli."
Udendo ciò, l'uomo fu Illuminato.

2 Una poesia... matematica.

Aleph-null bottles of beer on the wall,
Aleph-null bottles of beer,
You take one down, and pass it around,
Aleph-null bottles of beer on the wall.

$\aleph_0$ bottiglie di birra al muro,
$\aleph_0$ bottiglie di birra,
Ne prendi una, e la passi in giro,
$\aleph_0$ bottiglie di birra al muro.

Figura 4: : Ecco come appare il listato dell'esempio, dopo essere passato attraverso TEX.

Un'ultima nota: i caratteri **'$'** delimitano una formula matematica, e qui vengono usati per produrre il simbolo (``aleph con 0''). $\aleph_0$

Installazione

Se tutto questo vi ha solleticato, e volete provare LATEX sul campo, non vi resta che **installarlo**. La distribuzione di TEX più diffusa sotto Linux si chiama **teTEX**, ed è disponibile come **pacchetto precompilato** per Debian GNU/Linux, Red Hat Linux e S.u.S.E. Linux.
Alternativamente, potrete trovare teTEX presso:

http://www.tug.org/tetex/
ftp://ftp.dante.de/pub/tex/systems/unix/teTeX

ed ogni altro mirror del CTAN (Comprehensive TEX Archive Network).

Per saperne di piu'...

Il modo più rapido per imparare LATEX è senza dubbio ``*The Not So Short Introduction to LATEX , or LATEX in 87 minutes*'', che troverete (dopo aver installato teTEX) nel file */usr/lib/texmf/doc/latex/generic/lshort2e.dvi*. Questo libro spiega in 5 capitoli l'essenziale per scrivere in LATEX, anche se lascia spesso dei dubbi sulla precisa sintassi dei vari comandi.
Il riferimento sicuro per imparare LATEX è il libro di Leslie Lamport, ``*LATEX - A Document Preparation System*'', edito da Addison-Wesley: questo libro copre tutto quanto è necessario sapere per scrivere correntemente in LATEX, ma manca degli argomenti più avanzati e di spiegazioni su come piegare LATEX alle vostre esigenze se dovete andare oltre i formati standard.
Infine, teTEX viene con una documentazione molto completa, nei formati .dvi, .ps e .html; ogni pacchetto incluso in teTEX ha il suo manuale, che si può consultare dalla shell con **texdoc nome-del-pacchetto**.



INTRODUZIONE AL GCC

the gnu c compiler

FELICE MAINOLFI
f.mainolfi@oltrelinux.com

25 anni, laureando in Ingegneria Elettronica, si occupa di sistemi distribuiti e sicurezza in ambiente unix.

Per chi inizia con Linux, magari provenendo dal mondo Windows e senza avere una minima esperienza nel campo della programmazione, il primo incontro col compilatore puo' essere traumatico. In questo articolo delineeremo quali sono le caratteristiche salienti del compilatore e dei tool di sviluppo di GNU, senza tralasciare la pratica.

In sistemi operativi come **linux** il **compilatore** c/c++ ricopre un ruolo **fondamentale**: la maggior parte dei programmi è fornita come **sorgenti** e per utilizzarli è necessario **compilarli**. La filosofia free di linux vede nel compilatore c/c++ della GNU la scelta più naturale.
In questo articolo cercheremo di focalizzare quelle che sono le caratteristiche più importanti del compilatore e dei tools di programmazione GNU.

Il processo di compilazione

L'intero **processo** di compilazione richiede più **passi** come schematicamente riportato in figura 1. Si parte naturalmente dal **sorgente** .c o .cpp il quale viene passato ad un programma detto **preprocessore** che si incarica di **espandere** le apposite direttive. È in questo punto che vengono trattati i famosi **#include** e le **direttive** di compilazione **condizionale**. Una volta espanse le direttive del preprocessore la parola passa al compilatore vero e proprio. Il codice **binario eseguibile** non è prodotto direttamente; piuttosto viene sfornato il compilato in **assembler** (cioè un file di testo contenente il listato assembler equivalente al nostro programma c/c++). Il codice assembler viene passato a sua volta allo GNU assembler (as) che produce finalmente il **file oggetto**.
Perchè un file oggetto (.o) e non **direttamente** il binario eseguibile? Fino al momento in cui i nostri programmi risiedono in un unico file .c in effetti la cosa risulta **superflua**. Tuttavia è d'uso comune separare il proprio codice in una serie di moduli, oGNUno contenente funzioni che rispondono a compiti **logicamente** distinti. È normale che un file .c faccia riferimento a variabili o funzioni contenute in un altro file. Il compilatore ha la possibilità di trattare tali file, marcando come "**indefiniti**" tutti i nomi che **non** riesce a risolvere. Quello che si ottiene è una serie di file oggetto che contengono un certo numero di simboli non definiti; in altri termini si dice che i file non sono "**linkati**".
Il *deus ex machina* della situazione è il **linker**, un programma che preleva il **codice** e i **simboli** dai vari file **oggetto**, eventualmente **risolve** i nomi indefiniti e infine produce il file binario **eseguibile**.

Il ruolo di Gcc

Per nostra fortuna le varie fasi della compilazione vengono "**coordinate**" dal programma gcc che è parte integrante del GNU c. Gcc si incarica di **interpretare** la **linea di comando** fornita e decide i passi necessari alla **produzione** dell'eseguibile. Con opportuni parametri è possibile **bloccare** il processo in una fase qualsiasi. In tabella 1 trovate tutte le estensioni riconosciute da gcc e le azioni corrispondenti.

Un po' di pratica

Cominciamo con la chiamata "**classica**" del compilatore: abbiamo un unico file .c e vogliamo l'eseguibile. Se il sorgente è nel file programma.c il comando:

```
gcc -o pippo programma.c
```

produce **l'eseguibile** e lo memorizza nel file **pippo**. Possiamo osservare l'intero andamento del processo di compilazione attivando la modalità "**verbose**":

```
gcc -v -o pippo programma.c
```

Una possibilità interessante è quella di avere in uscita il programma tradotto in **assembler**:



Figura 1: Il sito http://www.GNU.org

```
gcc -S programma.c
```

troverete il codice nel file programma **.s** . Chi ha già programmato in assembler in ambienti Microsoft può trovarsi spiazzato leggendo il **codice** prodotto: niente paura, tipicamente il GNU assembler utilizza la cosidetta sintassi **at&t** che prevede **l'ordinamento inverso** degli operandi **sorgente** e **destinazione** e l'indicazione **esplicita** della lunghezza degli stessi, un pò come accade per i processori della famiglia 68000.

Vediamo ora con un piccolo **esempio** come è possibile compilare un programma organizzato su più file. Supponiamo di dover **calcolare** la somma di due numeri e per far questo costruiamo una **routine apposita** (listato 1) che metteremo nel file sommafun.c; possiamo compilare il file, ma dobbiamo fermarci al file **oggetto**: per ottenere un eseguibile è necessario definire la funzione **main()** cioè il punto di entrata del programma. Il file oggetto sommafun.o si ottiene con

```
gcc -c sommafun.c
```

A questo punto possiamo compilare il listato 2: esso è il programma che **sfrutterà** la funzione somma():

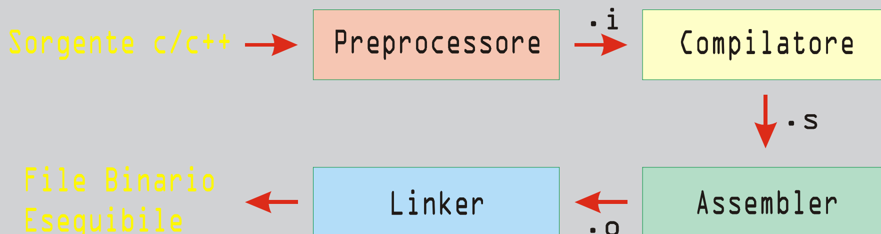
```
gcc -c progsomma.c
```

disponiamo ora dei due file progsomma.o e sommafun.o da linkare per produrre l'eseguibile:

```
gcc -o somma sommafun.o
                progsomma.o
```

Si noti che non sono necessari tutti questi passaggi.

figura 2



Lo stesso risultato si ottiene direttamente con il comando:

```
gcc -o somma sommafun.c
                progsomma.c
```

dove l'ordine dei file è inessenziale.

L'ottimizzazione

Una caratteristica molto interessante è la possibilità di **abilitare** varie **ottimizzazioni** durante la compilazione: questo può produrre un codice più **snello** e **veloce** ma in qualche caso particolare può **interferire** con la **corretta** compilazione. Per evitare problemi si può effettuare una "*regolazione fine*" del processo di ottimizzazione mediante l'uso sulla linea di comando di uno o più parametri **-f**. Purtroppo esistono più di una ventina di parametri -f dedicati all'ottimizzazione ma, fortunatamente, è possibile sfruttare schemi "standard" utilizzando i parametri -O e -O2. Provate a compilare il vostro programma nei due modi seguenti:

```
gcc -S programma.c
```

```
gcc -O -S programma.c
```

e date un'occhiata ai file .s che ottenete... una bella differenza, no?

Tabella 1

Estensioni	Azioni
.c	Sorgente C: preprocessa, compila, assembla
.C	Sorgente C++: preprocessa, compila, assembla
.cc	Sorgente C++: preprocessa, compila, assembla
.cxx	Sorgente C++: preprocessa, compila, assembla
.i, .ii	Sorgente C/C++ preprocessato: compila, assembla
.s	Sorgente assembler: assembla
.S	Sorgente assembler: preprocessa, assembla
.o	File oggetto: passato direttamente al linker
.a	File archivio: passato direttamente al linker

Listati

LISTATO 1
(file sommafun.c)

```
int somma (int a, int b)
{
  return (a+b);
}
```

LISTATO 2
(file progsomma.c)

```
#include <stdio.h>

void main(void)
{
  int sum;

  sum = somma (1, 5);
  printf(" somma(): %d\n", sum);
}
```

LISTATO 3

```
#include <stdio.h>
#include <math.h>

float calcola(float a, float b)
{
   return (sin((b*3)+a/20));
}

int main(int argc, char **argv)
{
   float calc = 0;
   unsigned int i;

   for (i = 0; i <= 500000; i++)
     calc = calcola(calc, 5.5);
}
```





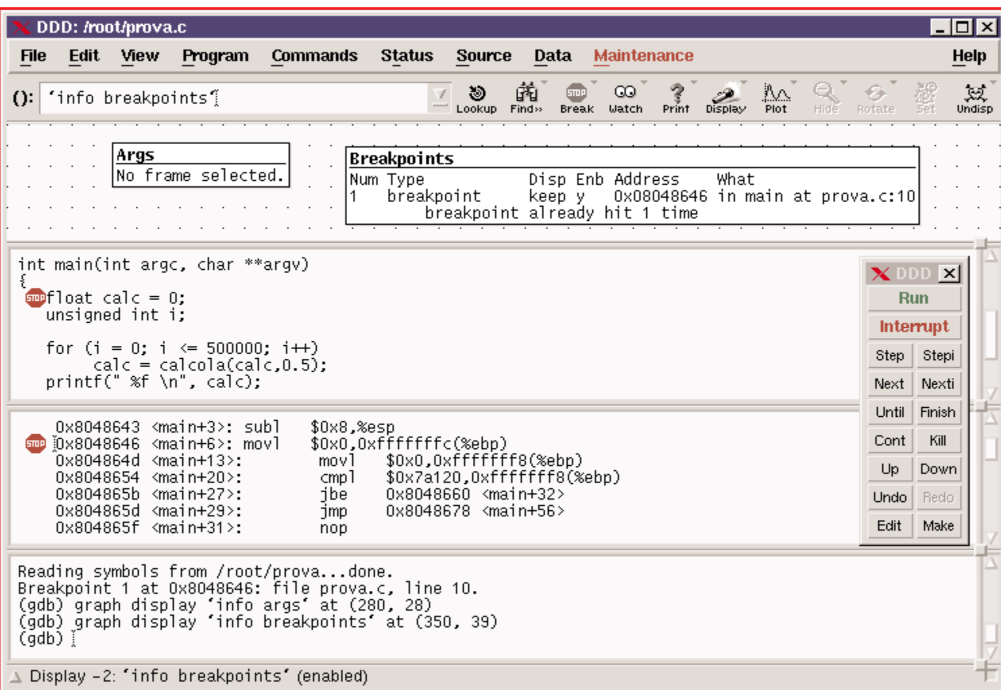


Figura 3: ddd, un eccellente front-end per gdb.

Il debugging

Per "debugging" si intende genericamente il **processo di analisi e correzione degli errori contenuti in un programma**. La GNU fornisce un "debugger" che aiuta a scovare gli errori eseguendo **passo passo** le istruzioni del programma e consentendo di **esaminare** diverse informazioni, come il **contenuto delle variabili o la gerarchia delle chiamate.**

Affinchè ciò sia possibile, il file **eseguibile** deve contenere anche una serie di informazioni **aggiuntive** che permettono al debugger di **associare** il codice macchina eseguito a una linea del sorgente e di sapere a quale "*simbolo*" è associata una particolare area di memoria. Naturalmente l'introduzioni di queste info produce un aumento delle **dimensioni** del programma eseguibile: una volta eliminati tutti gli **errori** sarà possibile **ricompilare** il programma **senza** queste informazioni.

L'introduzioni delle info per il debugging è possibile utilizzando il parametro **-g**. Esistono diverse **varianti** per questo parametro, ma illustrare il loro comportamento sarà oggetto di un prossimo articolo. Provate questi due comandi:

```
gcc -o pippo programma.c

gcc -g -o pippo programma.c
```

e osservate le dimensioni del file pippo prodotto ogni volta.

Gdb è il debugger della GNU. La sintassi di chiamata è molto semplice:

```
gdb programma
```

apparirà il prompt in attesa di comandi. Ad esempio potremo chiedere il **disassemblato** della main() con il comando:

```
disassemble main
```

oppure **stoppare** l'esecuzione impartendo un **breakpoint** su una funzione con il comando **break** e successivamente avviando il programma con **run**. L'uso di gdb, in definitiva, non è molto semplice e sarà oggetto di una serie di articoli. Tuttavia vi segnalo **ddd**, un front-end per **X** che rende l'uso di gdb comodo ed amichevole (vedi fig. 3). Potete prelevarlo, e ve lo consiglio caldamente, all'url http://www.cs.tu-bs.de/softech/ddd/ oppure nella sezione products di http://www.dislessici.org .

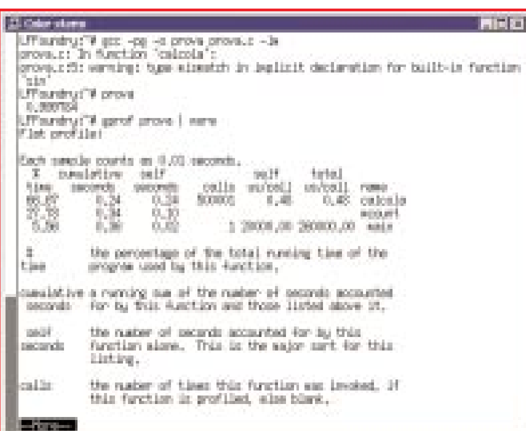


Figura 4: Il processo di profiling per il listato 1.

Profiling

Il profiling di un programma è **l'analisi del suo funzionamento** al fine di individuare le routine che **pesano** di più sul **tempo di esecuzione**; in questo modo è possibile **intervenire** laddove il codice si presenta troppo **lento** oppure viene eseguito più **spesso**.

Esaminiamo il programma del listato 2; non fa nulla di straordinario: si limita a chiamare **500mila** volte la routine calcola(). Chiamiamo il file sorgente provagp.c e compiliamolo con il comando:

```
gcc -lm -pg -o provagp
                    provagp.c
```

il parametro -pg induce il compilatore ad introdurre un pò di codice aggiuntivo. Se, infatti, avviamo provagp verrà prodotto il file **gmon.out** che contiene una serie di dati **utili** al profiling. Gmon.out può essere esaminato utilizzando il programma gprof (GNU profiler) :

```
gprof provagp | more
```

ottenendo un rapporto completo sul tempo **impiegato** da ogni **funzione** e dal numero di volte che essa è stata **chiamata** (vedi fig.4). Vengono inoltre forniti altri dati di tipo **statistico** come il tempo cumulato e la granularità, il tutto accompagnato da una legenda esauriente.

Approfondimenti

Lo spazio di questa introduzione non ci consente di approfondire altri aspetti. Tuttavia chiunque è interessato ad avere una visione più completa delle possibilità offerte dal compilatore GNU può fare riferimento al **gcc-howto** reperibile in italiano sul sito del PLUTO (http://pluto.linux.it) oppure scaricare il manuale completo in inglese direttamente da http://www.GNU.org .

Nelle prossime puntate...

Nei prossimi numeri, quando inizieremo con la programmazione **vera** e propria, avremo modo di sviscerare altri "**segreti**" del compilatore. Inoltre parleremo in maniera approfondita dei **tool di sviluppo come gli ambienti integrati**, il debugger gdb, il make e le tecniche per usarli al meglio.





EVOLUZIONE DELLA SPECIE

HTML  XML

JILANI KHALDI

j.khaldi@oltrelinux.com

Sviluppa prevalentemente in C/C++ e Delphi.
Si occupa di C/S, Calcolo Strutturale, CAD e grafica 3D (OpenGL).

XML come evoluzione dell'HTML. L'HyperText Markup Language fu inventato molti anni fa, quando neppure si sognava uno sviluppo della rete quale adesso è. Risente quindi di tutta una serie di «vizi» di progettazione che sono stati in parte superati con il nuovo XML. Inizieremo a conoscere questa nuova sintassi, sicuramente più chiara e precisa.

Secondo una delle leggi evoluzionistiche nella natura, ogni fenomeno che abbia raggiunto la piena maturità, **entra in crisi** e comincia a far intravedere i suoi propri **limiti**, per **evolvere** in un altro **fenomeno**, oppure **uscire** di scena e far parte della memoria storica. Tutto dipende ovviamente dalla natura e dalla portata del fenomeno. Rimanendo nel campo dinamico ed attuale dell'informatica, possiamo dire che la crisi del DOS ci ha condotto a Windows, quella di HTML a XML, e magari quella di Windows ci porterà a Linux. Questo solo per fare qualche esempio.

HTML e XML: origine comune

HTML (*HyperText Markup Language*), non è **veramente** un **linguaggio** di **programmazione** come potrebbe suggerire il nome, tipo il C oppure il Pascal. Da questo punto di vista non lo è neppure XML (*eXtensible Markup Language*), ma neanche SGML (*Standard Generalized Markup Language*) **padre** di tutti i due. HTML è semplicemente un **linguaggio di descrizione di pagina**. Cioè, indica solamente al programma che lo carica **come visualizzare** il testo contenuto in esso.

Ricordiamo però, che a differenza dei suoi derivati, SGML è un **meta-linguaggio** (*un linguaggio adatto a definire altri linguaggi*) molto **complesso** e gigantesco. Nell'ottica del programmatore, è comunque più corretto pensare a XML come SGML-- e non come HTML++; visto che a detta dei suoi autori, XML **racchiude 80%** della potenza di SGML.

HTML è un linguaggio soprattutto **semplice**; talmente semplice che ha reso la pubblicazione sul Web accessibile a tutti; anche a persone che non hanno precedenti conoscenze nel campo della programmazione e nemmeno la conoscenza del linguaggio HTML stesso. Il contenuto e il **successo** del Web sono cresciuti enormemente in questi anni proprio grazie a HTML.

La **crisi** di **HTML**, se così la possiamo chiamare, è in parte attribuibile alle nostre **continue pretese** a far **svolgere** ad esso compiti non perfettamente adatti, ed in parte a causa della sua **sintassi**, che essendo basata su un numero **finito** di **tag**, rimane rigorosamente fissa.

> «XML è nato proprio per risolvere alcuni problemi dell'HTML. In particolare si possono aggiungere nuovi tag, da noi definiti.»

Una delle grosse lacune di HTML è la sua scarsa **flessibilità** a descrivere la grande varietà di dati nelle pubblicazioni sul Web. Infatti, dispone solo dei tag **<HEAD>**, cioè intestazione, e **<BODY>**, cioè corpo, che, come suggerisce il nome, è un calderone all'interno del quale vengono inseriti praticamente tutte le altre informazioni. Ma se noi vogliamo strutturare i nostri dati e descriverli per esempio in capitoli, bibliografie, riferimenti... come facciamo?

XML è nato proprio per risolvere questo problema ed altri. La parola "**eXtensible**", cioè estendibile, sta a significare che possiamo **aggiungere** nuovi tag, **da noi definiti**, per descrivere qualsiasi tipo di informazione, come vedremo in seguito.

Per evidenziare qualche lacuna di HTML, basta questo semplice esempio:

```
<B> Questa linea viene visualizzata
in grassetto </B>
<I> E questa in corsivi </I)
<CENTER> E questa giustificata
centralmente </CENTER>
```

La prima riga verrà stampata in **grassetto**, la seconda in **corsivo**, e la terza sarà **giustificata** centralmente. Tutti questi tag modificano le proprietà di rappresentazione del testo, ma senza dire nulla sul suo **contenuto**.

Consideriamo quest'altro frammento di testo HTML:

```
<B>Attenzione! Salvate spesso i
vostri dati. Specialmente se lavorate
con Windows.</B>
```

Un programma di indicizzazione per esempio, non sarà in grado di ricavare nessun genere di informazione utile dal frammento precedente.

Lo stesso frammento scritto in XML diventa:

```
<NOTE
CLASS="CONSIGLI">Attenzione!
Salvate spesso i vostri dati.
Specialmente se lavorate con
Windows.</NOTE>
```

In questo modo è molto facile per il programma reperire l'informazione (che rappresenta un consiglio) tra **<NOTE>** e **</NOTE>**.

XML in pratica

Chi conosce e scrive in HTML non solo troverà XML altrettanto **semplice** ed **efficace**, ma soprattutto più **rigoroso** ed **elegante**, permettendogli di realizzare e facilmente mantenere con molto ordine, siti Web di notevoli dimensioni e varietà di contenuti.

Ora creiamo un altro frammento XML che rappresenti un lettera di invito, con nuovi dati inventati da noi a proposito:

```
<LETTERA>
<DATA ALIGN="RIGHT">
15 luglio 1999
</DATA>
<INDIRIZZO>
Antares Informatica <BR/>
Via M. della Libertà 5 <BR/>
66054 Vasto (CH) <BR/>
</INDIRIZZO>
<CORTESE_ATTENZIONE>
Responsabile sistemi
informativi
</CORTESE_ATTENZIONE>
<OGGETTO>
Invito alla partecipazione ad
una festa di inaugurazione
</OGGETTO>
<BODY>
<P ALIGN="JUSTIFY">
Abbiamo il piacere di
invitarLa alla nostra festa
che si terrà il giorno 10
settembre, alle ore 20.30,
presso l'Hotel Rimini. La
preghiamo di non mancare.
</P>
<P>
Cordiali saluti.
</P>
</BODY>
<FIRMA>
Dr. Pasquale Lofestaiolo
</FIRMA>
</LETTERA>
```



Figura 1: Il sito del W3C a www.w3c.org, dove è possibile scaricare Amaya.

Analizzando il contenuto del frammento, notiamo che è stata posta l'enfasi sul **tipo** di informazioni che contiene, piuttosto che sulle modalità di rappresentazione delle stesse. A parte il tag <P> e quello di allineamento <ALIGN>.

La **libertà** con la quale si possono **definire nuovi tipi di dati**, è una dei maggiori punti di forza di XML. Il problema ora da risolvere è quello di trovare una maniera semplice ed efficace, per permettere ai campi DATA, INDIRIZZO, OGGETTO…, di essere visibili e comprensibili ad una applicazione esterna. Qui entra in scena il **DTD**.

Document Type Definition

Ossia DTD, **rappresenta un insieme di regole per descrivere i dati definiti dall'utente all'interno del documento XML**. Tale insieme indica per ogni dato i suoi attributi, i valori degli attributi e la gerarchia all'interno degli stessi, visto che alcuni dati possono anche contenere altri. Diciamo una specie di header file, per intenderci.

Il DTD della lettera presentata sarà il seguente:

```
<!ELEMENT lettera (data,
indirizzo, cortese_attenzione,
oggetto, body, firma?)>

<!ELEMENT data (#PCDATA)>

<!ATTLIST data align (left |
right) "left")>

<!ELEMENT indirizzo (#PCDATA |
lv*)>

<!ELEMENT lv EMPTY>

<!ELEMENT cortese_attenzione
(#PCDATA)>

<!ELEMENT oggetto (#PCDATA)>

<!ELEMENT body (p+)>

<!ELEMENT p (#PCDATA)>

<!ATTLIST p align (left |
justify | right) "left")>

<!ELEMENT firma (#PCDATA)>
```

Ecco la **descrizione** di tutte le **informazioni** contenute in ogni riga:

1- *indica che una lettera è composta da questi dati: data, indirizzo, cortese_attenzione, oggetto, body, e firma.*
2- *Una data contiene solo testo.*
3- *Una data può essere giustificata a sinistra, oppure a destra. Giustificazione di default a sinistra.*
4- *Un indirizzo contiene un testo, e può contenere una linea vuota, più di una, o nessuna.*
5- *Una linea vuota non contiene nessun dato.*
6- *Il nome di chi riceve la lettera è un testo.*
7- *L'oggetto della lettera è un testo.*
8- *Il contenuto della lettera può contenere più paragrafi.*
9- *Ogni paragrafo contiene solo testo.*
10- *Un paragrafo può essere giustificata a sinistra, oppure a destra. Giustificazione di default a sinistra.*
11- *Una lettera finisce con una firma.*

A questo punto, appare chiaro come è stata facilmente creata una corrispondenza tra i dati all'interno del documento e la loro descrizione tramite il DTD, per poterli gestire in seguito da applicazioni esterne, con tutti i vantaggi che ne derivano.
XML rappresenta senz'altro un **notevole passo avanti** rispetto a HTML. E' uno strumento molto elegante e sofisticato per pubblicare dati complessi su Internet ed Intranet aziendali. Possono essere facilmente rappresentati: formule di matematica, di chimica, disegni vettoriali, note musicali... XML ha tutte le carte in regola per diventare lo **standard** di fatto non solo per la pubblicazione sul Web, ma anche per proporsi come **formato universale** per la **rappresentazione** dei dati, risolvendo quel cronico problema di incompatibilità tra formati generati da pac-



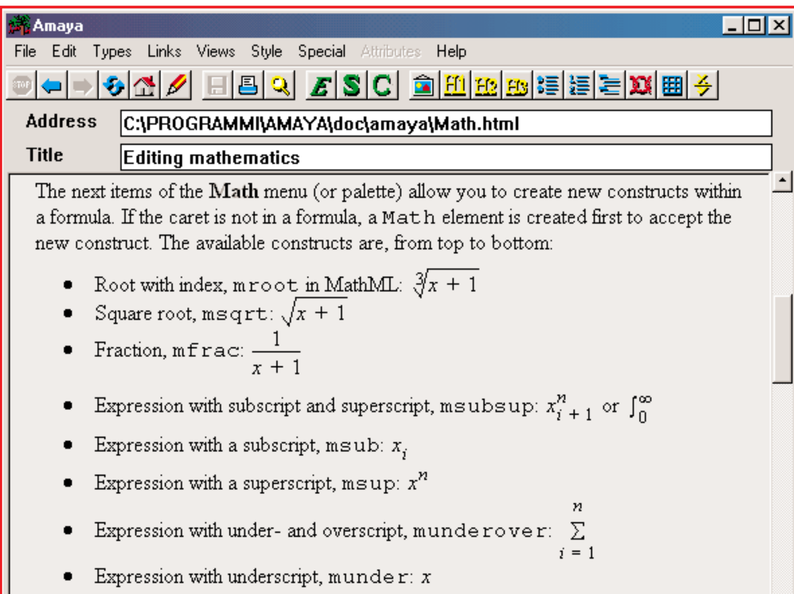


Figura 2: Amaya, il browser sperimentale del W3C, è molto adatto per pubblicare testi scientifici sul WEB.

chetti diversi. Lascio a voi immaginare i benefici che si otterranno quando tutti i file generati da Microsoft Office, Corel Office ..., avranno il formato nativo XML.

Situazione attuale dei browser

Il **W3C** (*World Wide Web Consortium*) che rappresenta **l'ente de jure** a **standardizzare** le nuove tecnologie per il Web, fa fatica a soddisfare tutte le nuove e continue richieste di **estensioni** a **standard** già stabiliti ed a **imporre la sua autorità** per evitare la **frammentazione** di queste **tecnologie**. La contesa del mercato dei browser tra Microsoft e Netscape ha **vanificato** il processo di standardizzazione e ha condotto ad estensioni proprietarie ai rispettivi browser, rendendoli **parzialmente** incompatibili tra loro. Non è infatti raro imbattersi in siti con suggerimenti tipo questi: Best Viewed by Internet Explorer; oppure, Best Viewed by Netscape Navigator. I problemi maggiori non riguardano solo chi naviga, ma soprattutto chi crea e mantiene siti Web, **costretto** a scrivere codice **interpretabile** a seconda del browser in uso, o saggiamente, limitarsi al loro **minimo** comune multiplo.
Con l'estensione dei browser attuali con parser XML, non credo affatto che il problema verrà risolto. La ragione è sempre la stessa. I contendenti, Microsoft e Netscape, continueranno a cercare di **dominare** il mercato, invece di **servirlo**.
La situazione attuale vede solo **IE 5** XML enabled, ma con **riserva**; visto che si limita a **tradurre** il codice XML all'interno del documento nel rispettivo **HTML prima** di essere visualizzato. Questo mi fa ricordare i primi "**finti**" compilatori C++; erano semplici compilatori C, che traducevano il codice C++ in C prima di compilarlo. A parte IE c'è l'ottimo **Amaya**, il browser sperimentale del W3C, **disponibile con codice sorgente** sia per **UNIX** che per **Windows**. Lo potete trovare sul sito del W3C: http://www.w3.org. Amaya è contemporaneamente **Web browser** e **Web editor**; molto snello e adatto a quegli utenti che possiedono computer poco potenti. Ha un proprio linguaggio macro per essere personalizzato ed esteso con nuove funzionalità; molto adatto per le **Intranet aziendali** e nelle **scuole**, dove si continua ancora ad usare **vecchi 486**. E' molto adatto per pubblicare **testi scientifici** sul Web (vedi figura2).

Editor XML disponibili

Per scrivere codice XML, **teoricamente** basta qualsiasi **elaboratore di testi**. Ma se il documento diventa complesso, allora conviene rivolgersi ad editor **specializzati**. Uno di questi è senz'altro XML Pro della Vervet Logic (http://www.vervet.com), caratterizzato dalla **semplicità** di **utilizzo** e le numerose funzionalità. In un futuro molto prossimo, vedremo tutti gli editor HTML di oggi evolvere nei rispettivi editor XML di domani. Ancora solo un po' di pazienza.

E' finita per HTML?

Niente affatto. Anzi, HTML continuerà a **prosperare** e rappresentare lo strumento principe di pubblicazione sul Web per i prossimi anni. Ogni linguaggio di programmazione, o qualsiasi altro strumento informatico, è una **soluzione**, appositamente creata per risolvere un **determinato problema**. **Tale soluzione continuerà ad esistere, finché la causa che l'ha generata permane.**

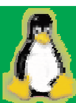
Riferimenti:

http://www.w3.org



HTML	XML
Caratteristica principale	**Caratteristica principale**
Linguaggio di marcatura fissa di pagine Web	Linguaggio di marcatura logica ed astratta di documenti
Pro	**Pro**
Semplicità	Molto adatto per la pubblicazione di dati complessi
Molto ben conosciuto e diffuso	Ha una sintassi molto pulita e rigorosa
Disponibilità di molti strumenti di sviluppo	Adatto per diventare uno standard per la rappresentazione di molti tipi dati
Interpretabile da tutti i browser attuali	Un motore di ricerca può facilmente trovare dati contenuti in un documento XML
Contro	**Contro**
Non è adatto per la pubblicazione di dati complessi	Complessità
Il documento tende a diventare caotico al crescere delle sue dimensioni	Ridondanza
Non ha una sintassi rigorosa	Poca disponibilità di strumenti di sviluppo
	Non è per ora interpretabile da tutti i browser attuali

Tabella 1: Breve comparazione tra XTML e XML



Eventi Linux

Silvia Samarro - carina -

HACK-IT NOI C'ERAVAMO

la manifestazione tutta italiana riguardante il mondo dell'hacking!

h99 hacaro

con la testa un po' meno :-)

Allarme tra gli studiosi di comportamento animale: una colonia della temutissima e tuttora poco conosciuta specie "Hacarus Informaticus" , radunatasi attraverso tutta la penisola per mezzo di sotterranei contatti, ha portato con la sua presenza, scalpore e preoccupazione negli ambienti scientifici.
L'esemplare di Hacarus, solitamente dedito ad attività informatiche oscure e potenzialmente insidiose, deregolato e naturalmente portato a pericolose sperimentazioni, adoratore del dio pinguino simbolo di libertà e fedeltà, è considerato da più parti dannoso ed infestante e quindi tenuto il più possibile sotto stretto controllo, nelle sue rarissime apparizioni spontanee.
La straordinaria migrazione si è verificata durante un tranquillo fine settimana di metà giugno. Centro del raduno, che si ha motivo di temere ritualmente annuale, uno stabile dimesso, nella periferia milanese. Cauti osservatori hanno potuto, mimetizzandosi efficacemente, assistere da vicino e registrare dinamiche e schemi dell'attività comunitaria, allo scopo di diffonderne la conoscenza circa aspetto ed abitudini, per la salvaguardia di ogni cittadino e la protezione del comune capitale informatico.
Monitor nello zainetto, cd e floppy nelle tasche anteriori, keyboard sotto la t-shirt e scanner opzionale nella tasca posteriore: in questo modo l'Hacarus dissimula il suo agguerrito potenziale.
Dotato di forte senso della territorialità, si dimostra particolarmente aggressivo verso i propri simili durante la delicata fase di "installazione". Infatti è di interesse vitale per l'esemplare poter alloggiare in modo ottimale case e tastiera: il rituale prevede oltre alla tempestiva e strategica occupazione di ripiano e sedile, un minuzioso ed elaborato lavoro di stesura degli svariati, chilometrici cavi in dotazione, e del loro tortuoso collegamento alle prese di rete.
Nello svolgimento delle proprie attività, l'Hacarus Informaticus ha sviluppato cicli di veglia lunghi fino a ventidue ore, durante i quali opera con frenesia, saldamente ancorato alla tastiera, in modalità solo apparentemente autistica, ma incurante di condizioni climatiche, necessita' fisiologiche ed eventi circostanti.
Le sue straordinarie doti di resistenza ed adattabilità gli hanno consentito infatti di appisolarsi senza problemi solo per pochi minuti, sdraiato sui materassini polverosi della palestrina o sprofondato nell'unico divanetto disponibile del bar al primo piano, durante la punta massima della "sosta margaritas" del sabato sera.
La cura parentale presso l'Hacarus Informaticus prende forma di trasmissione del bagaglio conoscitivo: si è dimostrato infatti massimo lo sforzo della comunità nell'organizzare un flusso costante ed omogeneo di nozioni teoriche e pratiche, strumento indispensabile alla sopravvivenza stessa dell'esemplare.
Ma su quali siano i meccanismi riproduttivi dell' Hacarus, ci si interroga tuttora:
sebbene infatti sia stata avvistata una sorpren-

Identikit di un hackaro

Nickname:
Variano moltissimo: in crescita **Neo** e **Morpheus** dopo il film "Matrix".

Domicilio abituale:
Non ha una dimora fissa: con particolari cadenze emigra per incontrare i suoi simili.

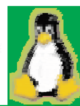
Rapporti con l'altro sesso:
Risale ad un'era geologicamente non datata la sua ultima conquista.

Film preferiti:
WarGames gli ha cambiato la vita. "Matrix", visto solo 0010 0110 volte.

Segni particolari:
Ha almeno una foto in camera con la tornta in faccia a Bill Gates.



Figura1: Strani individui frequentavano l'hack-it... le loro foto segnaletiche sono state prontamente inviate alla pubblica sicurezza. Sullo sfondo, la centrale operativa montata a tempo di record.



dente percentuale di esemplari femmina, la stessa non risulterebbe sufficiente ad un'efficace salvaguardia della specie, e l'argomento rimane al centro del dibattito scientifico.
Una rivoluzionaria teoria parla di "contagio simpatico": basterebbe infatti un solo esemplare di Hacarus Informaticus sufficientemente entusiasta e aperto al dialogo, per spingere alla trasformazione "n" esemplari di altre specie. La mutazione così avviata sarebbe quindi rapida, devastante, e purtroppo, irreversibile.
Per la sicurezza nazionale, enti senza scopo di lucro come Telecom, ed organizzazioni neutrali su modello Poltel, hanno messo spontaneamente a disposizione osservatori speciali e studiosi del fenomeno, che continueranno anche in seguito una delicata opera di monitoraggio degli esemplari catalogati.
Durante tutto il protrarsi dell'evento, comunque, bollettini quotidiani delle più autorevoli testate hanno largamente pubblicizzato il fenomeno, mettendo in allarme la popolazione, invitando alla prudenza e sconsigliando il minimo contatto con la specie.

www.ecn.org/hackit99

channel IRC ecn.org:6667 #hackit99

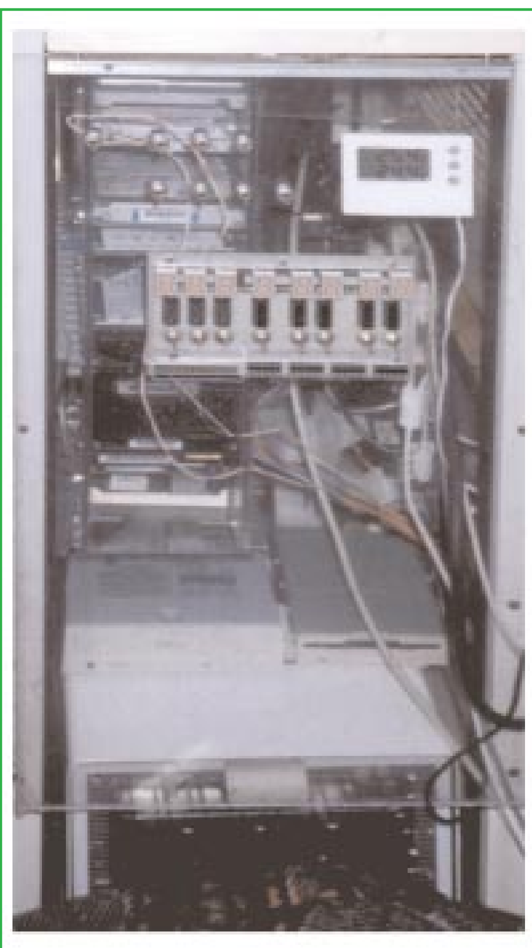


Figura2: Uno dei computer in sede all'hack-it... Da notare, in basso, il fornetto per scaldare le merendine, controllato - ovviamente - dal pc.

L'Hackit va, l'HackLab resta.

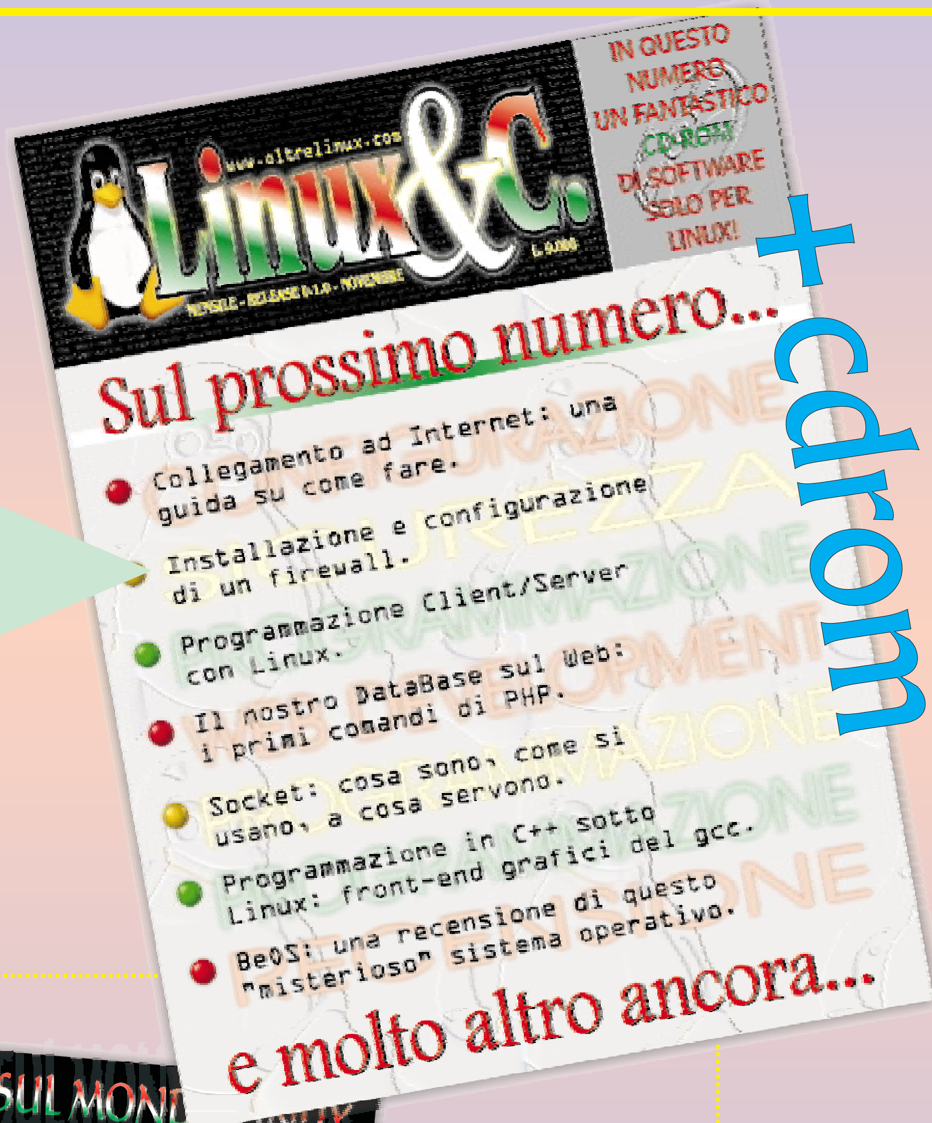
Incubi notturni? Delirio e sudori freddi durante il giorno? Settantadue ore di furore all'anno non vi bastano e languite nell'attesa del prossimo raduno... tranquilli, la crisi d'astinenza da H-it è un problema risolto. Smanettoni già nostalgici ed entusiasti dell'ultima ora possono trovare l'ambiente del raduno in pillola mensile o settimanale nei "laboratori hackit" delle principali città: Firenze (www.firenze.linux.it/~leandro/hacklab/), Catania (www.kyuzz.org/freaknet), Roma (www.ecn.org/forte/forthnet) e ancora Savona, Milano, Napoli, Torino e Bologna quelli già attivi o in partenza. Per tutti sperimentazione, formazione ed informazione all'insegna del free software e della comunione dell'hardware (svuotate gli armadi!!! L' H-Lab è la tana giusta per pezzi vecchi ma ancora efficienti).

Rete e Diritti Civili

28 Agosto 1999, Carlo Gubitosa presenta il suo "Italian crack down" alla Casa dei Pensieri presso la Festa dell'Unità di Bologna.
Accanto a lui, anche Andrea Monti e Stefano Chiccarelli di "Spaghetti hacker", Gianfranco Livraghi di "Alcei" (Associazione per la libertà nella comunicazione elettronica interattiva: www.alcei.it) e Nicola Rabbi autore di "Handicap in rete", per fare il punto su "stili di vita, democrazia, poteri, internet".
Italian crack down ripercorre la nascita della rete, da Arpanet alle BBS o "Bacheche Elettroniche", per raccontare l'operazione Hardware 1 che nel maggio 1994 diede il via in Italia ad una campagna di oscuramento e sequestro a tappeto delle reti telematiche autogestite. Irruzioni, perquisizioni, avvisi di garanzia, sequestro indiscriminato di ogni apparecchiatura che all'epoca sconvolsero sysop e semplici utenti, nella maggior parte dei casi determinarono, oltre a perdite finanziarie e danni d'immagine, la chiusura dei nodi "incriminati" e nutrirono la stampa con una senzionalistica quanto infondata caccia al pirata informatico.
Dopo più di cinque anni, cercando di fare chiarezza nelle dinamiche che scatenarono la vicenda, sul fenomeno crack down si allunga l'ombra della BSA (Business Software Alliance, che dalla fine degli anni '80 raggruppa sei tra i maggiori produttori di software del mondo). La BSA promosse infatti in Italia tra il 1992 e l'anno successivo una campagna supportata da numero verde, per segnalare fruitori di software abusivo, i quali vennero successivamente denunciati alla magistratura.
Sostenitrice di una particolarissima interpretazione del diritto d'autore, che renderebbe perseguibile anche la singola copia priva di regolare licenza d'uso installata ad esempio dal professore sul computer scolastico, la BSA ha potuto, attraverso l'immagine di autorità competente, manovrare nel proprio interesse il clima di vuoto legislativo e l'ingenuità tecnica di procure e forze dell'ordine per fare piazza pulita del fenomeno del software abusivo ad uso personale.
Gubitosa pubblica a fondo libro l'interessante Manifesto per la libertà della comunicazione (<documento in evoluzione continua>) : il pieno diritto alla comunicazione si afferma anche attraverso la libertà di stampa e il diritto di "accesso", la tutela della privacy, il rispetto della giustizia ed il rifiuto della censura, la denuncia della concentrazione dei poteri informativi ed il riconoscimento dell'utilità pubblica delle BBS.
Italian crack down e' anche on-line, all'indirizzo ww.apogeonline.com/openpress/libri/529 e attenzione, una sicura novità : via libera a riproduzioni e trasmissioni dell'opera sotto qualsiasi forma e con qualsiasi mezzo (salvi naturalmente gli scopi commerciali...) perchè, per stessa ammissione dell'autore, fotocopie citazioni ed estratti, rappresentando pubblicità ulteriore, non costituiscono minaccia o lesione dei suoi diritti. Maggiori approfondimenti su libertà, diritti ed indipendenza dell'informazione in rete nei prossimi numeri.

Vi abbiamo offerto il numero 0 OMAGGIO.

Un altro **REGALO** Vi abbiamo riservato per il **PRIMO** numero.





Un fantastico **MOUSEPAD** con il calendario del **2000**!

In edicola il 15 di ottobre. Non mancate.